EFFECT OF DISTURBANCE DYNAMIC CHARACTERISTICS ON OPTIMUM PID CONTROLLER TUNING CONSTANTS

By

SIRIPOL KOSINSANI

Bachelor of Science
Kasetsart University
Bangkok, Thailand
1973

Master of Agriculture
Mississippi State University
Mississippi State, Mississippi
1976

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 1985

EFFECT OF DISTURBANCE DYNAMIC CHARACTERISTICS ON
OPTIMUM PID CONTROLLER TUNING CONSTANTS

Thesis Approved:

Archibald D. Hill
Thesis Adviser

Bennett Basore

Marvin L. Stone

W. Morel

Norman N. Durham
Dean of the Graduate College

ACKNOWLEDGMENTS

I am especially indebted to the chairman of my advisory committee, Dr. Bennett L. Basore (Elec. Eng.) for his permission for me to participate in the Ph.D. program in the School of General Engineering and his concern with problems facing me during my stay at Oklahoma State University. I wish to express my warmest appreciation to my thesis adviser, Dr. Archibald G. Hill (Chem. Eng.) for his encouragement and valuable guidance. Special gratitude is due to the other committee members, Dr. Peter M. Moretti (Mech. Eng.) and Dr. Marvin L. Stone (Agric. Eng.) for their intelligent advisement and kindness.

I am also thankful to Dr. John P. Chandler (Com. Sci.) for his advisement on fitting curves to the data.

As forevermore, I am grateful to my parents, my wife (Dr. Jarungsri Kosinsani, D.D.S.) for their love, understanding, and waiting of my return.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure | Page

Figure		Page

Figure		Page

Figure		Page

Figure		Page

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

The research work to be described by this dissertation deals with the problem of selecting tuning constants for feedback control systems. The control systems considered are those that involve a single controlled variable and a single manipulated variable. The controller considered is the conventional three mode proportional-integral-derivative (PID) controller. The research approach will be to use a digital computer to simulate the performance of a feedback control loop including: the process model, sensor, PID controller, and valve. For a particular set of tuning constants the response of the control system to a disturbance will be evaluated using a figure of merit based on the integral criteria known as the "integral of the absolute value of the error (IAE)."

The optimum tuning constants (giving the lowest IAE) are dependent upon the parameters of the control loop dynamic model and the disturbance dynamic model. Previous workers have dealt with this problem using disturbance dynamic models based on simple step changes in set-point and load variables, and a sequence of random step changes in the load variable. The unique feature of this research will be to extend the investigation of the effect of the disturbance dynamic model by using a first-order model for the disturbance.

Similar to the efforts of previous workers this research will deal with a generalized process based on the first-order plus dead time (FOPDT) mathematical model. For a particular set of conditions optimum tuning constants will be found using the control system model as the objective function of an optimization program suitable for a multivariable search involving a nonlinear function. Controller actions investigated will include both proportional-integral and proportional-integral-derivative.

The results of this research will be applicable to practical control problems such as heat exchanger control and distillation control. In these applications it is known that the controlled variable responds with different dynamics to changes in the disturbance variable and the manipulated variable. Control system performance based upon tuning constants found by the present research should be superior to tuning constants found by conventional methods that consider only the dynamic response of the controlled variable to changes in the manipulated variable.

CHAPTER II

LITERATURE REVIEW

The development of automatic control instrumentation suitable for use in the chemical process industries created an incentive for engineers to examine the dynamic performance of process equipment. Process variables such as temperature, pressure, flow, and level which had previously been controlled manually could now potentially be controlled automatically by the new technology. A first step in the application of feedback control technology is to mathematically model the system to be controlled. Efforts to develop process models have proceeded along two lines. One approach is to consider the unsteady state material and energy balances involved with a particular process. This method supposes that it is possible to derive the differential equations, based on fundamental principles, that adequately describe the system performance. It is appropriate for simple systems such as blending in a stirred vessel, level control, etc. An alternative approach has been used for more complex processes such as heat exchanger control or distillation control. In these systems a fundamental derivation would require complex mathematics such as partial differential equations. The alternative approach assumes that the physical process to be controlled is available for dynamic testing. Tests are performed to examine the dynamic response of a controlled variable to a change in some manipulated variable. The process response is then fitted to a less complex model based on ordinary differential equations. Use of this technique introduces some error due to approximation and is generally valid only for small changes in the process variable.

The general process models most often used to fit data from dynamic tests include the first order plus dead time model and the second order plus dead time model given below in transfer function notation:

$$\text{FOPDT} \quad G_p(s) = \frac{K_p e^{-\theta_d s}}{\tau s + 1} \qquad [1]$$

$$\text{SOPDT} \quad G_p(s) = \frac{K_p e^{-\theta_d s}}{\tau^2 s^2 + 2\zeta \tau s + 1} \qquad [2]$$

Procedures to fit experimental data to the above models have been reported by Zoss [20], Anderson [1], Sten [18], and Deshpande and Ash [5].

The three mode proportional-integral-derivative controller which first gained acceptance after World War II is today still the most frequently applied controller. Mathematical descriptions are given below:

$$\text{time domain} \quad V(t) = K_C \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t)\, dt + \tau_D \frac{de(t)}{dt}\right] + V_S \qquad [3]$$

$$\text{Laplace domain} \quad \frac{V(s)}{E(s)} = G_C(s) = K_C\left[1 + \frac{1}{\tau_I s} + \tau_D s\right] \qquad [4]$$

Use of the PID controller involves the specification of tuning constants: K_C - proportional gain, τ_I - integral time and τ_D - derivative time. If the derivative time tuning constant is set equal to zero, then the controller reduces to the two mode proportional - integral (PI) controller.

Digital computer simulation is an effective way to determine the performance of a control system as it responds to a disturbance. Luyben [9] has given a number of examples of programs that simulate the components of a feedback control loop. In his text he describes how the Euler integration technique may be used to numerically solve a set of differential equations representing a control system. In digital simulations two forms of the PID controller are often used [5]. These include the position and velocity forms given below:

Position Form:

$$V_n = V_o + K_C \left[E_n + \frac{T}{\tau_I} \sum_{i=o}^{n} E_i + \frac{\tau_D}{T} (E_n - E_{n-1})\right] \qquad [5]$$

Velocity Form:

$$V_n = V_{n-1} + K_C \left[(E_n - E_{n-1}) + \frac{T}{\tau_I} E_n + \frac{\tau_D}{T} (E_n - 2E_{n-1} + E_{n-2})\right] \qquad [6]$$

In the above equations the subscript n refers to the nth sampling instant. The velocity form is obtained from the position form by subtracting $V_n - V_{n-1}$. When the above equations are used to simulate a continuous controller the sampling time T is set equal to the Euler integration time interval.

Selection of tuning constants for a control system may be accomplished using a trial-and-error procedure when a digital computer is available for simulation of process response. The integral of the absolute value of the error (IAE) criterion could be used as a basis for comparison. A set of starting values for the tuning constants could be found using a conventional tuning constant correlation given in the literature. Among the first correlations to be developed were those reported by Ziegler and Nichols [19] and Cohen and Coon [4]. These early methods were semi-empirical in nature and related to the stability considerations found in linear control theory. More recent correlations have been developed with the aid of the digital computer. Lopez et al. [8] developed correlations to find the optimum tuning constants for systems responding to step changes in load. Rovira et al. [13] performed a similar study for systems responding to step changes in set point.

Lopez and Rovira worked with process models including FOPDT and SOPDT. Their control systems could be represented by the diagram shown in Figure 1. It should be noted that this diagram assumes that the controlled variable C(s) responds to a change in load L(s) or manipulated variable V(s) with the same dynamic response represented as $G_p(s)$.

Lopez and Rovira both developed their correlations by using digital simulations of control systems in combination with an optimization program. The optimization program may be considered a formalized trial-and-error procedure. The objective of the optimization program was to find tuning constants that gave the minimum value of an integral criteria such as the IAE. These workers used an optimization program such as the technique described by Rosenbrock [11].

Sood and Huddleston [16, 17] described a study in which they used an optimization procedure to find tuning constants for systems exposed to a sequence of step load changes of random magnitude. An interesting discovery reported by these workers was the presence of local minimums in the IAE for tuning constant values outside of the range predicted by previously developed correlations. In some cases these unexpected local minimums proved to be global minimums. This study emphasizes the need to consider several starting values for tuning constants when using a unimodal optimization technique.

CHAPTER III

CALCULATION OF OPTIMUM TUNING CONSTANTS

Research Objectives

Digital simulation of feedback control systems has been used by previous workers to find optimum PID controller tuning constants. These previous studies have evaluated the effect of the process model as seen by the controller but have been limited to narrowly defined disturbances. The primary purpose of this research is to systematically study the effect of disturbance dynamics on optimum PID controller tuning constants. The criterion of performance will be to find tuning constants that minimize the integral of the absolute value of the error. Previous workers have developed correlations that give the optimum tuning constants as a function of process parameters. In this work the optimum tuning constants will be found as a function of both process and disturbance dynamic characteristics. While using the integral of the absolute value of the error as an optimization criterion, previous workers have not reported this value as a function of the process parameters in a manner similar to the correlations for tuning constants. A secondary objective of this research will be to provide such correlations for the integral of the absolute value of the error. This type of correlation can be used to illustrate the effect of process dead time and disturbance dynamics on control system performance. A third objective of this study will be to evaluate the effect of disturbance magnitude. If the feedback control system is modeled as a system of linear equations without constraints, the magnitude of the disturbance will linearly affect the integral of the absolute value of the error but will not affect the values of the optimum tuning constants. However, when constraints are added, such as upper and lower limits on the manipulated variable corresponding to an actual valve, disturbances with magnitudes large enough to saturate the valve will affect the optimum tuning constants calculation. Tuning constants reported in this study will be calculated for responses to disturbances that have magnitudes below that which give valve saturation. For a number of cases the magnitude of the disturbance that would saturate the valve is calculated.

Research Approach

The effect of disturbance dynamics may be studied using a feedback control system represented by the general block diagram shown in Figure 2. This control system differs from that used by previous workers shown in Figure 1. The difference involves the way in which the load L(s) enters the loop. The presently described system provides a separate transfer function $G_{p2}(s)$ to represent the dynamic effect of the load variable L(s) on the response of the controlled variable C(s). In this diagram the dynamic effect of the manipulated variable V(s) is represented by the transfer function $G_{p1}(s)$. In the control system studied by previous workers a single process transfer function $G_p(s)$ is provided to represent the dynamic effect of both the disturbance variable and the manipulated variable. The system shown in Figure 2 will reduce to that used by previous workers when the transfer functions $G_{p1}(s)$ and $G_{p2}(s)$ have the same form.

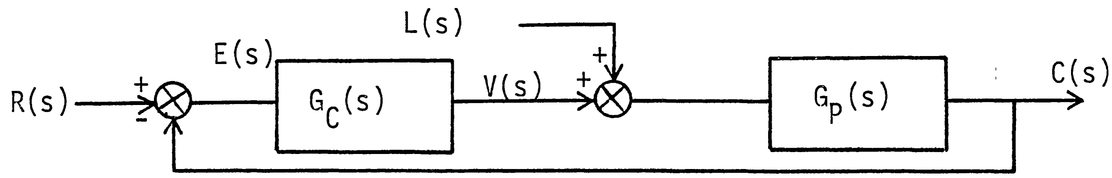


Figure 1. Block Diagram Used by Lopez and Rovira

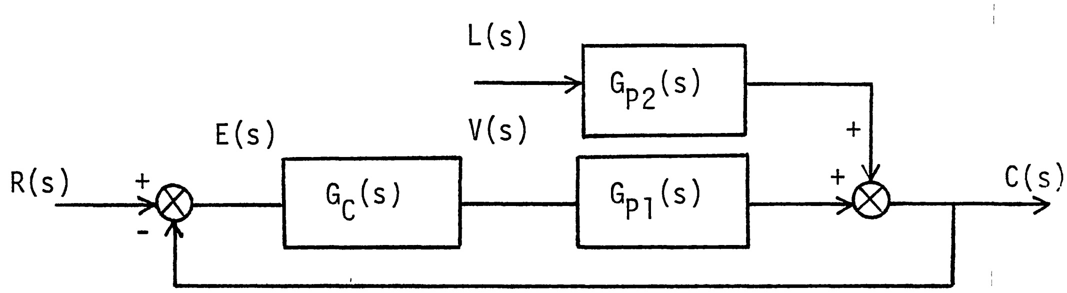


Figure 2. Block Diagram Used to Study the Effect of Disturbance Dynamics

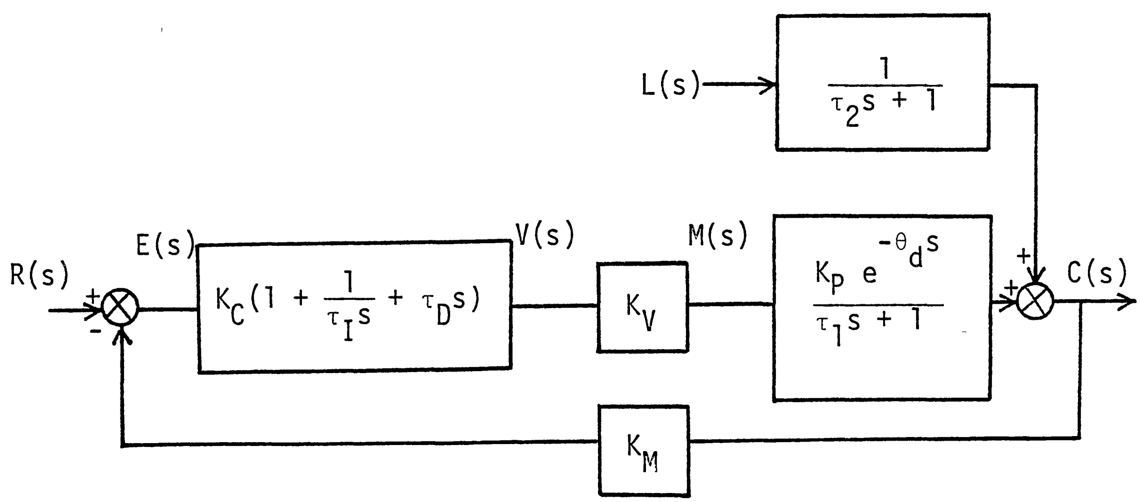


Figure 3. Control System With a FOPDT Process and a First Order Disturbance

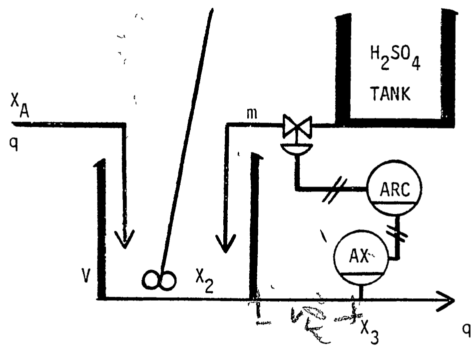
A more specific version of the control system described by Figure 2 is shown in Figure 3. In this system the transfer function giving the effect of the manipulated variable is shown as a first order plus dead time model with gain K_P. The transfer function giving the effect of the load variable is described as a simple first order process with unity gain. Gain terms have also been added for the measuring element K_M and the valve K_V. The controller is represented as an ideal PID controller having tuning constants K_C, τ_I, and τ_D. In the event that τ_D is set to zero the controller reduces to the two-mode PI controller. The research performed in this study will be applicable to a variety of different physical systems that may be represented by Figure 3.

The control system studied in this research has been given the particular physical interpretation shown in Figure 4. The process is depicted as a mixing tank with a constant liquid level and flow rate. The entering liquid, flowing at one liter per minute, has a nominal concentration of 500 mg/L of sulfuric acid (H_2SO_4). Concentrated sulfuric acid is added at a nominal mass flow rate m equal to 1000 mg/min. The flow of acid m is manipulated by a feedback controller in order to maintain an acid concentration of 1500 mg/L at the point in the exit liquid line where a concentration analyzer is positioned.

The concentration of the mixing tank is assumed to be homogeneous at all times due to perfect mixing. The liquid volume in the tank equals one liter providing a tank detention time of one minute. The liquid is assumed to obey ideal plug flow in the exit liquid line. The volume of the exit liquid line preceding the analyzer is allowed to vary in this study giving transportation lags in the range of 0.05 to 1.0 minute. The block diagram in Figure 4 shows three possible disturbances that may

Transportation Lag.

PROCESS DIAGRAM



NOT SHOWN ABOVE: X_1, X_B, X_4

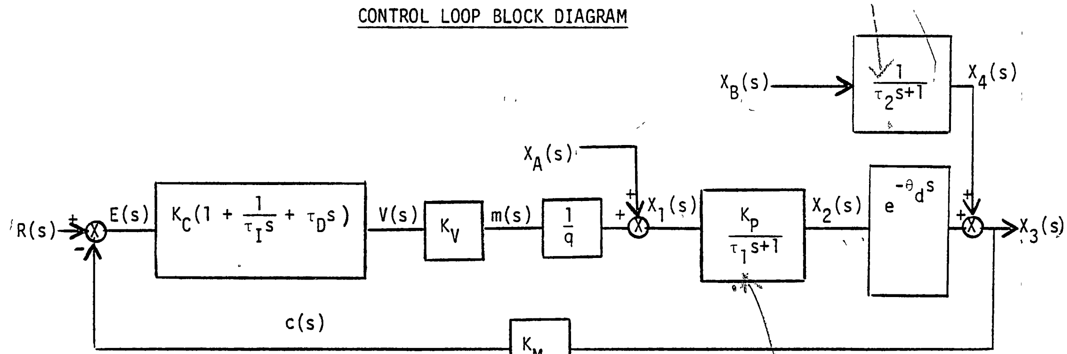
LEGEND

X_A = entering concentration, mg/L
X_2 = tank concentration, mg/L
X_3 = concentration measured by the analyzer, mg/L
m = acid flow, mg/min
V = tank volume = 1 liter
q = flow rate = 1 liter/min

INITIAL VALUES (NOT IN DEVIATION FORM)

X_A = 500 mg/L
X_2 = X_3 = 1500 mg/L
m = 1000 mg/min

CONTROL LOOP BLOCK DIAGRAM



ASSUME $X_1(s) \approx X_A(s) + \frac{m(s)}{q}$

$X_B(s)$ = step change in measurement error of the analyzer
$X_4(s)$ = measurement error of the analyzer after a 1st order lag

Figure 4. Physical Interpretation of a Control System Based on a FOPDT Process

enter the loop. These include a setpoint disturbance R, a load disturbance X_A, and a load disturbance X_B. The setpoint disturbance R corresponds to the type of disturbance studied by Rovira et al. [13]. The load disturbance X_A, physically interpreted as a step change in entering liquid concentration, corresponds to the type of disturbance studied by Lopez et al. [8]. The load disturbance X_B, may be physically interpreted as a step change in the measurement error of the analyzer. (This research is not limited to disturbances in measurement error; it is intended to apply to any first order disturbance). In an actual process this might be due to fouling of the sensor. The measurement error of the analyzer is considered to enter the loop as a first order transfer function. This transfer function corresponds to that represented as $G_{P2}(s)$ in Figure 2. In this study the first order time constant τ_2 associated with disturbance X_B was allowed to vary in comparison to the time constant τ_1 associated with the transfer function giving the dynamic response to the manipulated variable. The range of variation of the ratio of τ_2/τ_1 extended from 0.05 to 7.0.

The process diagram shown in Figure 4 illustrates the manipulated variable m as a mass flow rate of concentrated acid. Due to the low concentrations involved in this process the mass flow of acid is much less than the entering liquid flow. If the entering liquid is assumed to be water the actual ratio of mass flows of the two streams is less than 1/1000 at normal operating conditions. Under these circumstances the effect of the entering liquid concentration X_A and the acid mass flow rate m can be combined to give a hypothetical entering liquid concentration X_1 according to the following approximate equation:

$$X_1 = X_A + \frac{m}{q} \qquad [7]$$

where q is the constant liquid flow rate of one liter per minute and m is the acid mass flow rate in mg/min. The units of X_1 are mg/L. In this equation and all following equations used in the digital simulation of the mixing process, the variables are assumed to be deviations from steady state values. Therefore, at normal operating conditions, prior to the introduction of a disturbance, all variables would have a value equal to zero.

The process model as seen by the controller is represented by a first order plus dead time model relating the dynamic response of the concentration passing the analyzer to changes in the hypothetical concentration X_1 entering the tank. This model may be derived from a mass balance around the tank and a consideration of the transportation lag. The tank mass balance is:

$$\text{input} - \text{output} = \text{accumulation} \qquad [8]$$

$$X_1 \cdot q - X_2 \cdot q = V \cdot \frac{dX_2}{dt} \qquad [9]$$

- model for mixer

where:

X_1 = hypothetical entering concentration, mg/L

X_2 = concentration in the tank, mg/L

V = tank volume (one liter)

q = liquid flow rate (one liter per minute)

Rearranging gives:

$$\frac{dX_2}{dt} = \frac{X_1 - X_2}{\tau_1} \qquad [10]$$

where:

$$\tau_1 = V/q \qquad [11]$$

The transportation lag of the liquid passing from the tank to the analyzer may be calculated as:

$$\theta_d = \frac{V_L}{q} \qquad [12]$$

where:

θ_d = transportation lag (or dead time), minutes

V_L = liquid volume of the exit liquid line preceding the analyzer, liters

q = liquid flow rate (one liter per minute)

In the control system shown in Figure 4 the signals between the analyzer and the controller, and between the controller and the valve are depicted as pneumatic signals. Pneumatic control signals normally range from 3 to 15 psig. The gains associated with the measuring element K_M and the valve K_V were calculated as follows:

$$K_V = \frac{\Delta m}{\Delta V} = \frac{2000 \text{ mg/min}}{12 \text{ psi}} = 166.667 \qquad [13]$$

where:

Δm = maximum acid flow corresponding to maximum valve signal

ΔV = range of controller output signal, psi

$$K_M = \frac{\Delta c}{\Delta X} = \frac{12 \text{ psi}}{3000 \text{ mg/L}} = 0.004 \qquad [14]$$

where:

Δc = range of analyzer output signal, psi

ΔX = maximum concentration measured by the analyzer, mg/L

The research approach used in this study involved the digital simulation of the control system shown in Figure 4. The digital simulation of this control system was used as the objective function for an optimization program based on the Rosenbrock technique [6, 11]. Optimum tuning constants for step changes in X_B were found for various ratios of τ_2/τ_1 and θ_d/τ_1.

Description of the Objective Function

A program showing the way in which the previously described control system may be simulated digitally is illustrated in Figure 5. This program, written in UCSD Pascal [3], contains a function definition labeled as "function object." Function object calculates the integral of the absolute value of the error that accumulates as the system responds to a disturbance. In UCSD Pascal it is possible to assign global values to constants and variables that appear in a function definition. In this program the following constants and variables are assigned values globally with respect to the function definition: kmm, measuring element gain; kv, valve gain; r, step change in setpoint; xa, step change in entering liquid concentration; xb, step change in measurement error; delta, time step size for Euler integration; tt, total time of the disturbance response simulation; thetad, transportation lag or dead time; tau1, first order time constant for the response of the process to changes in the manipulated variable; tau2, first order time constant associated with load variable xb; s, integral number of time steps

```
PROGRAM MIXING-PROCESS;

const

   kmm=0.004;kv=166.666666667;

var

   x4dot,x2dot:real;
   c, {TRANSMITTED VARIABLE,PSI}
   er, {ERROR, CURRENT VALUE}
   epast, {ERROR, PREVIOUS ITERATION}
   erint, {TIME INTEGRAL OF ERROR}
   absie, {TIME INTEGRAL OF ABSOLUTE VALUE OF THE ERROR}
   x1, {xa+v*kv/1}
   x2, {PROCESS RESPONSE TO x1, BEFORE DEAD TIME}
   x4, {PROCESS RESPONSE TO xb}
   x3, {x2 AFTER DEAD TIME + x4, (THE CONTROLLED VARIABLE,MG/L)} time,
   vv, {CONTROLLER OUTPUT BEFORE CONSTRAINTS,PSI}
   va:real; {CONTROLLER OUTPUT AFTER CONSTRAINTS,PSI}
   h,g,q:integer; {POINTERS FOR DEAD TIME ARRAY}
   kc,taui,taud, {TUNING CONSTANTS}
   r, {STEP CHANGE IN SETPOINT,PSI}
   xa, {STEP CHANGE IN LOAD VARIABLE NO. 1,MG/L}
   xb, {STEP CHANGE IN LOAD VARIABLE NO. 2,MG/L
       (FOLLOWED BY 1ST ORDER DELAY TAU2)}
   delta, {ITERATION TIME INTERVAL}
   tau1, {1ST ORDER TIME CONSTANT FOR RESPONSE TO x1}
   tau2, {1ST ORDER TIME CONSTANT FOR RESPONSE TO xb}
   tt, {TOTAL TIME OF SIMULATION}
   thetad:real; {DEAD TIME}
   fname:string;
   printer:text;
   s:integer;{NO. OF ELEMENTS IN DEAD TIME ARRAY}
   sumn:real;
   dt:array[1..255] of real;

procedure data;

   begin
      readln(input,r,xa,xb);
      readln(input,delta,tt,thetad);
      readln(input,tau1,tau2,fname);
      s:=round(thetad/delta) + 1;

   end;
```

Figure 5. Pascal Program Illustrating Control System Simulation

```
procedure initial;

   begin
      writeln(printer,r,xa,xb);
      writeln(printer,delta,tt,thetad);
      writeln(printer,tau1,tau2,fname);
   end;

procedure answer;
   begin
      writeln(printer);
      writeln(printer,kc,taui);
      writeln(printer,' VALUE OF FUNCTION= ' ,sumn:16:8);
   end;

function object(kcc,tauii:real):real;

   begin
      c:=0.0;epast:=0.0;erint:=0.0;absie:=0.0;
      for h:= 1 to s do dt[h]:= 0.0;
      g:=s; q:=1;time:=0.0;x2:=0.0;x3:=0.0;x4:=0.0;
      while time<tt do
         begin
            c:=kmm*x3;
            er:=r-c;
            va:= kcc * (er + erint/tauii + (er - epast)*taud/delta );
            vv:= va;
            if va <= -6.0 then va:= -6.0;
            if va >=  6.0 then va:=  6.0;
            x1:= xa + va*kv;
            x4dot:= (xb - x4)/tau2;
            x2dot:= (x1 - x2)/tau1;
            time:= time + delta;
            if (vv < 6.1) and (vv > -6.1) then
               erint:= erint + er * delta;
            absie:= absie + abs(er * delta);
            epast:= er;
            x2:= x2 + x2dot * delta;
            dt[g]:=x2;
            x4:= x4 + x4dot * delta;
            x3:=dt[q] + x4;
            g:= g + 1 ; q:= q + 1;
            if g > s then g:= 1;
            if q > s then q:= 1;
         end;
      object:= absie;
   end;
```

Figure 5. (Continued)

```
begin {MAIN PROGRAM}

   data;
   rewrite(printer,fname);
   taud:=0.0;
   initial;
   s:= round(thetad/delta)+1;
   kc:=51.2;
   taui:=12.8;
   while kc >= 0.2 do
      begin
         while taui >= 0.05 do
            begin
               sumn:=object(kc,taui);
               answer;
               taui:= taui/2.0;
            end;
         kc:= kc/2.0;
         taui:= 12.8;
      end;

end.
```

Figure 5 (Continued)

(delta values) included in the dead time thetad; and taud, the derivative time tuning constant. Variables that may change values each time the function object is evaluated include kc, the proportional gain tuning constant, and taui, the integral time tuning constant. In the case of this program taud is set to zero reducing the PID controller to a proportional integral controller.

The function object begins by initializing several process variables to zero corresponding to the initial steady state condition prior to a disturbance. Recall that this simulation uses deviation variables throughout. The time response of the control system is simulated using the Euler integration technique [9]. A loop is used to repetitively perform several calculations for each step in time. The loop continues to evaluate these equations while the running value of time is less than or equal to the simulation time tt.

The calculations performed in the program loop follow an order similar to that shown in the block diagram of Figure 4. The order of calculation is as follows: c, analyzer transmitted signal; er, error entering the PID controller; vv, PID controller output prior to implementation of constraints; va, PID controller output constrained to the normal pneumatic signal range; x1, hypothetical entering liquid concentration; x4dot, x2dot, derivative values; time; erint, time integral of the error used in the controller calculation; absie, time integral of the absolute value of the error; x2, concentration in the tank; x4, measurement error after first order lag tau2; and x3, measured concentration passing the analyzer (after imposition of the measurement error). At the completion of the loop the function object is assigned a value equal to the last cumulated value of absie, the integral of the absolute value of the error.

Additional details concerning the control system simulation may be emphasized as follows. The PID controller algorithm used in function object is written in the position form. The variable erint is used to store the integral of the error. This term is allowed to accumulate as long as the computed valve position vv does not exceed the valve constraints by more than a small margin. The purpose of this logic is to eliminate the development of "reset windup." A major advantage of computer analysis of control system performance is the ease with which dead time may be simulated. In this program an array dt is set up to simulate the dead time. Process variable x2, corresponding to the mixing tank concentration, is held in array dt for s number of time increments, the total of which is equal to the dead time thetad. Array position pointers g and q keep track of the positions in the array to be entered and exited by the process variable as it would have entered and exited the pipeline leaving the tank.

In the program shown in Figure 5 function object is called by a main program that successively calculates the integral of the absolute value of the error for a wide variety of PI controller tuning constant pairs. For a particular program run the load disturbance remains the same while the tuning constants are varied. The purpose of this program is to generate enough values of function object so that a topographical map may be prepared showing the form of the response surface. The results of this program were entered into a commercial software package sold by SAS Institute [14] in order to prepare the contour plots shown in Figures 6 to 12.

These plots show how the normalized integral of the absolute value of the error varies as a function of the tuning constant values. They

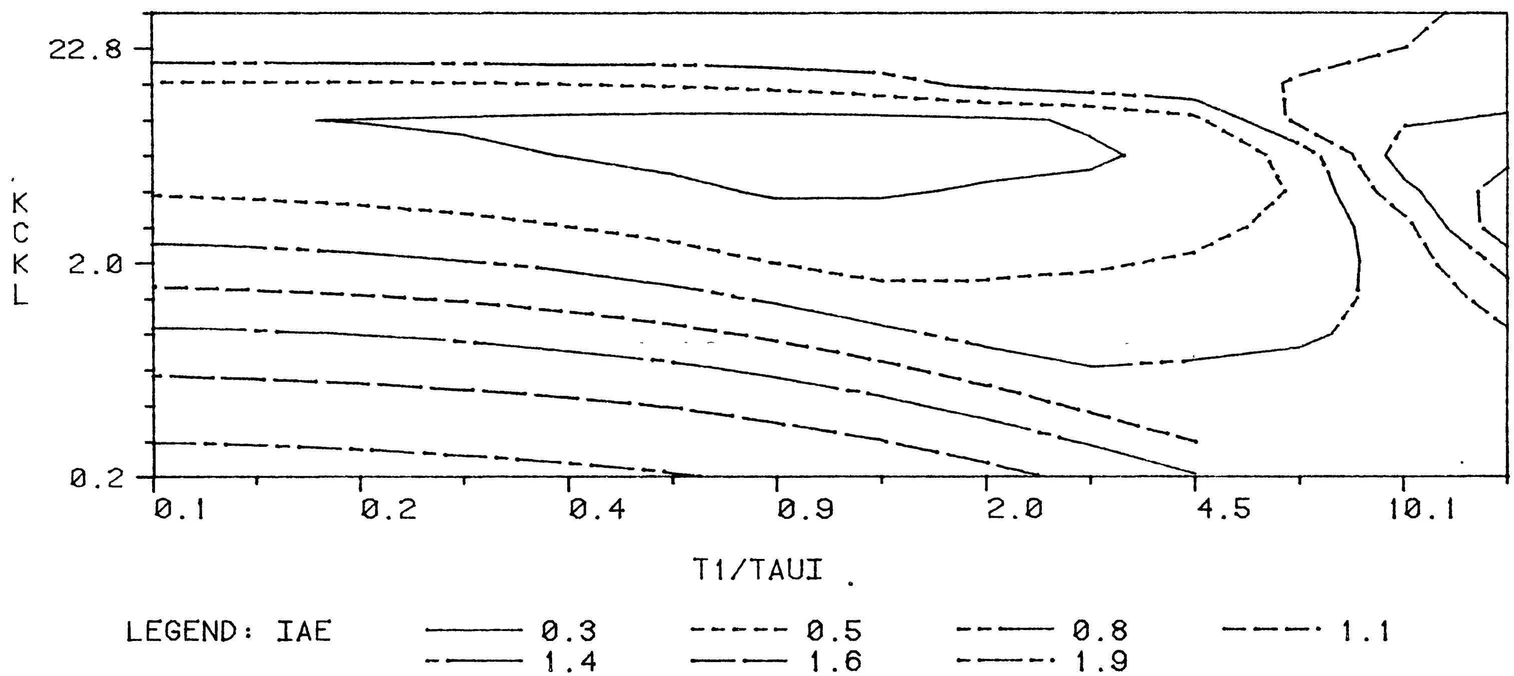


Figure 6. Effect of PI Controller Tuning Constants on the IAE for $\tau_2/\tau_1 = 0.1$

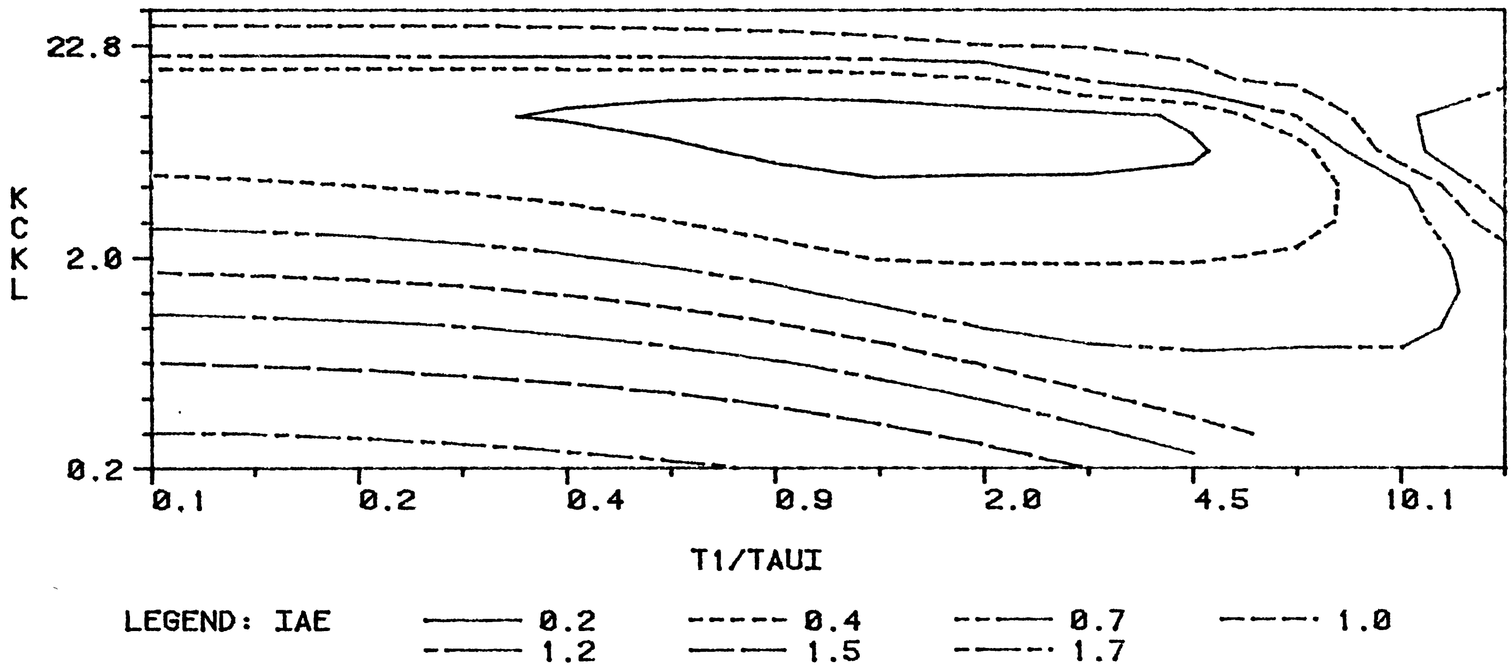


Figure 7. Effect of PI Controller Tuning Constants on the IAE for $\tau_2/\tau_1 = 0.3$

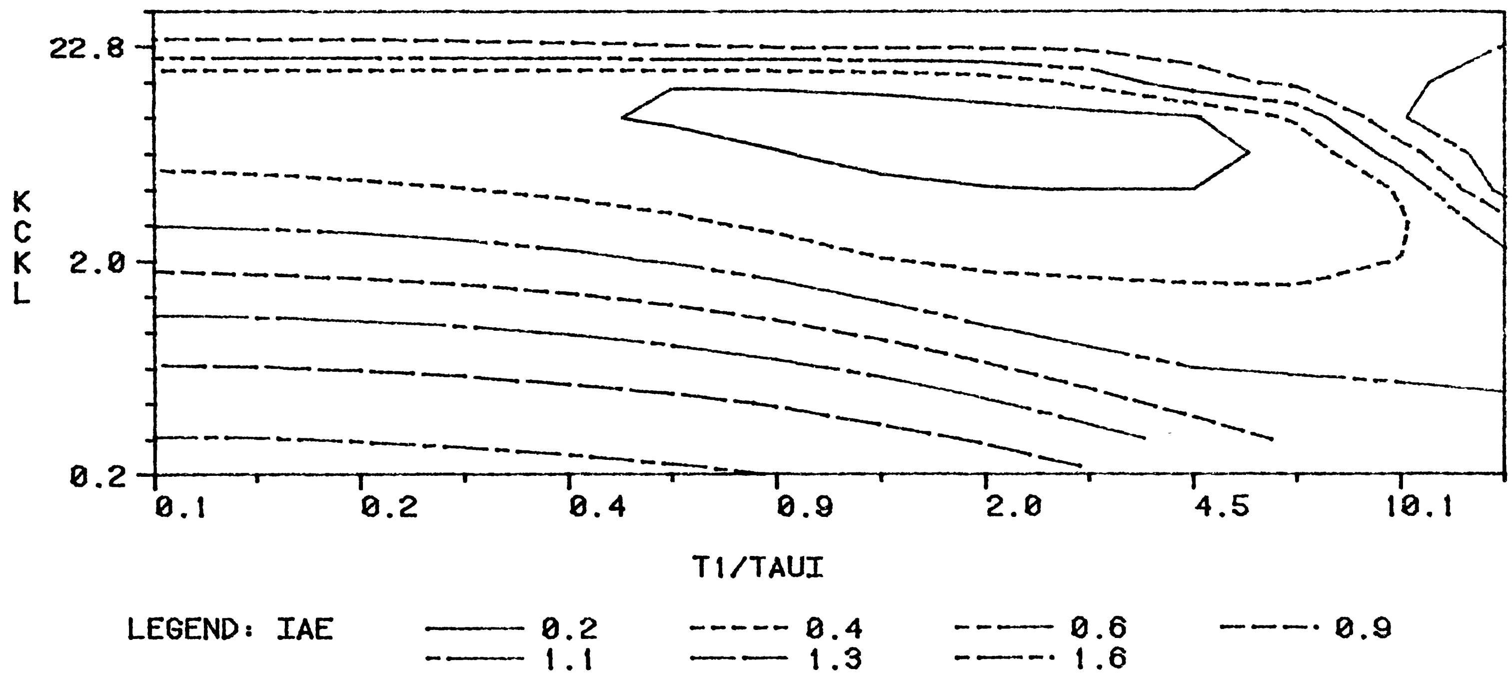


Figure 8. Effect of PI Controller Tuning Constants on the IAE for τ_2/τ_1 = 0.5

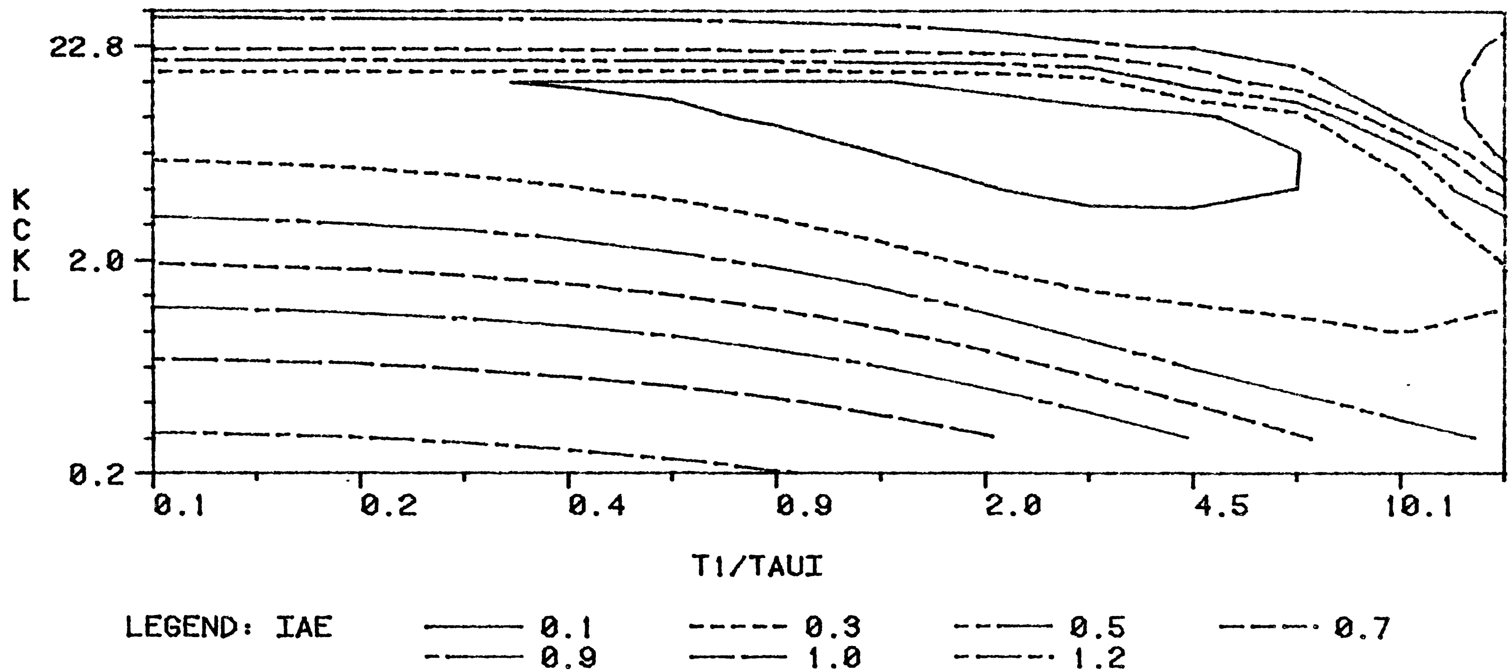


Figure 9. Effect of PI Controller Tuning Constants on the IAE for τ_2/τ_1 = 1.0

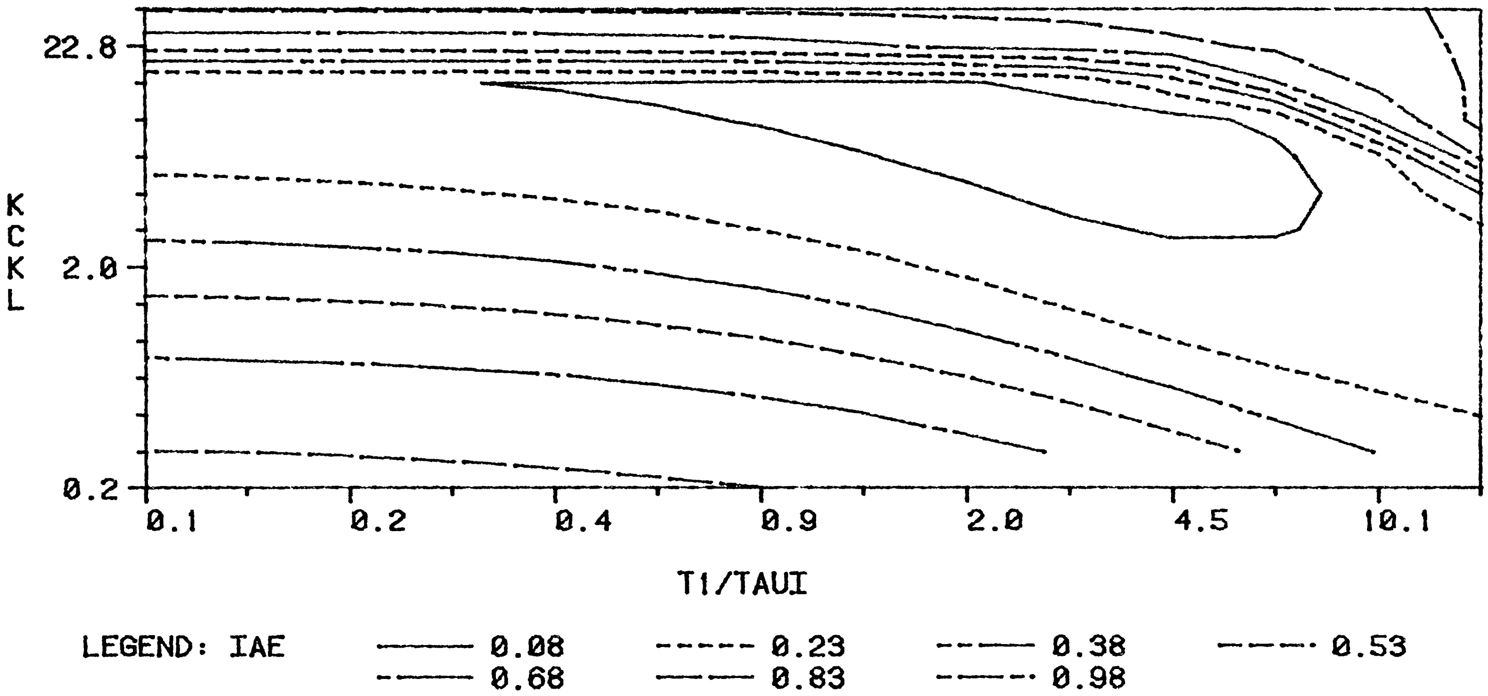


Figure 10. Effect of PI Controller Tuning Constants on the IAE for $\tau_2/\tau_1 = 2.0$

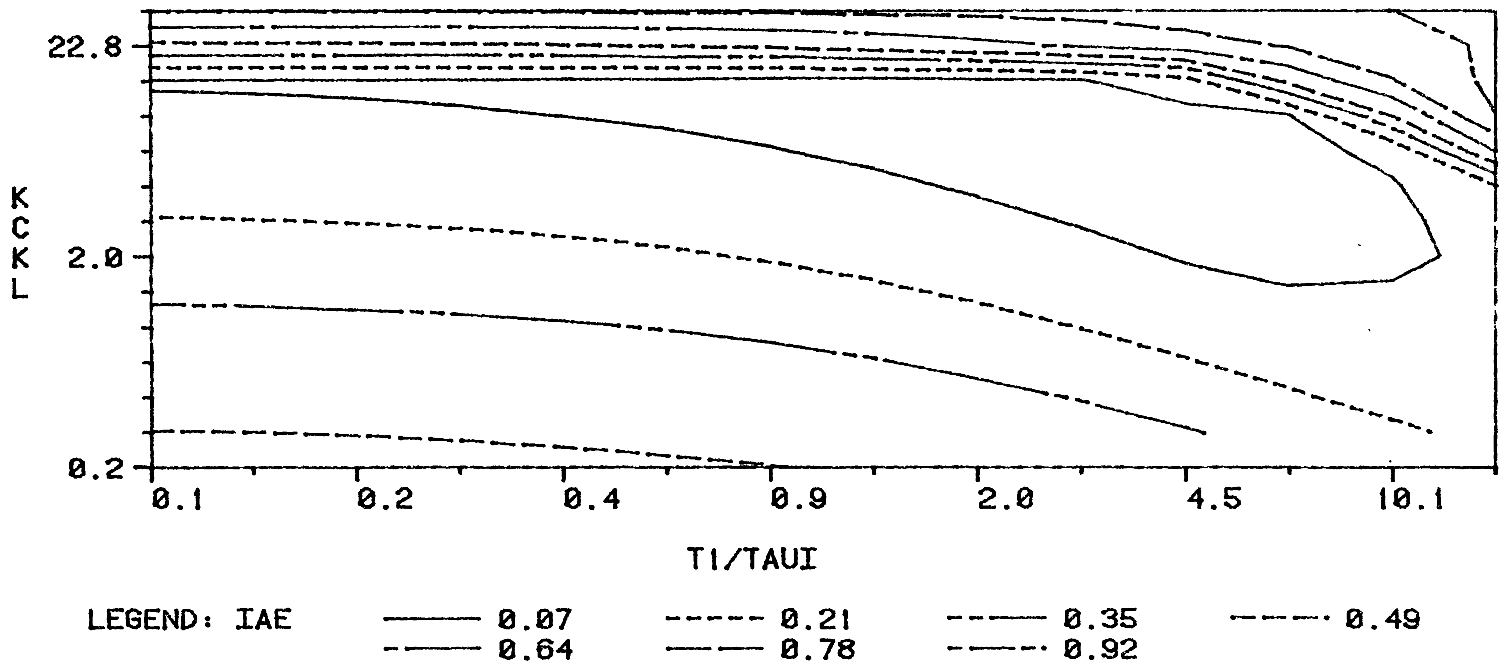


Figure 11. Effect of PI Controller Tuning Constants on the IAE for $\tau_2/\tau_1 \doteq 4.0$

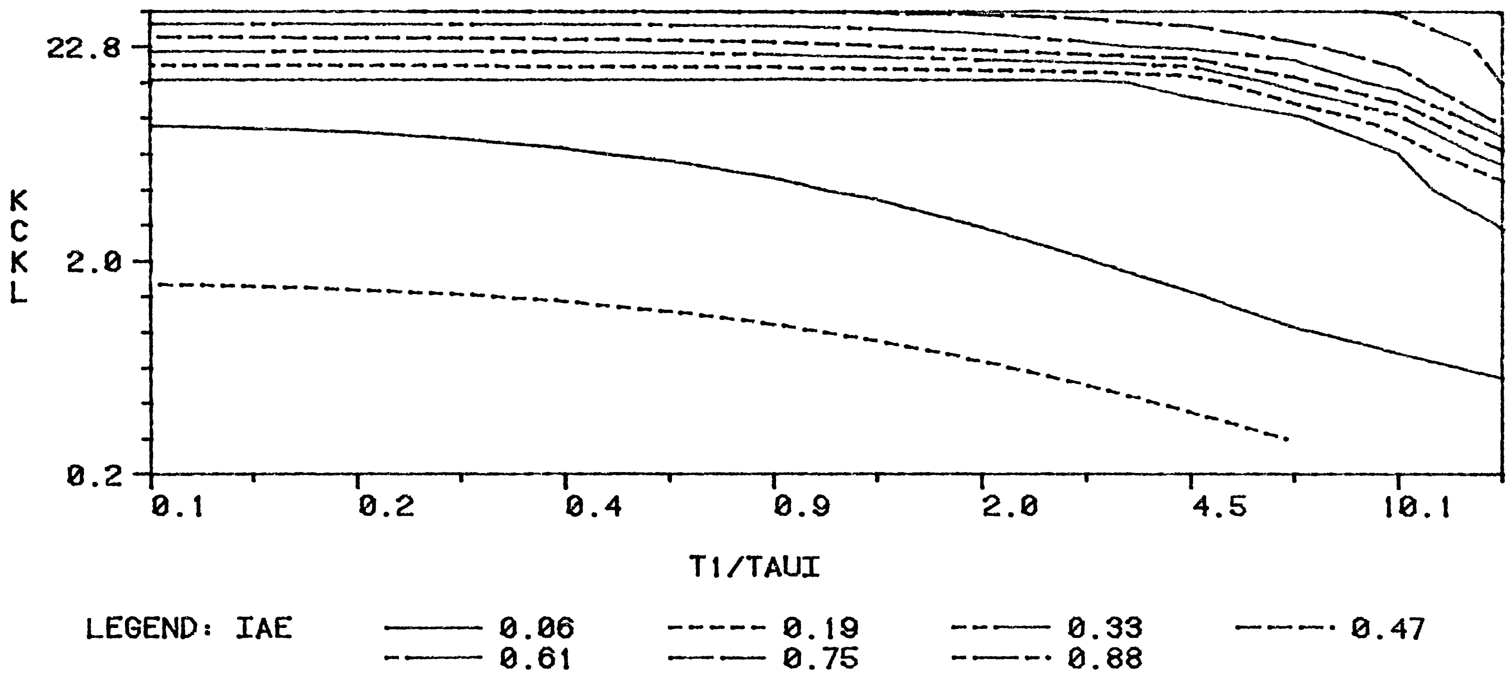


Figure 12. Effect of PI Controller Tuning Constants on the IAE for $\tau_2/\tau_1 = 7.0$

were prepared for the following conditions: r = 0, xa = 0, xb = 100, delta = 0.0005, tt = 2.31, thetad = 0.1, and tau1 = 1. The value of tau2 varied for each figure using the following values: tau2 = 0.1, 0.3, 0.5, 1, 2, 4, and 7. In order that the results from this system may be applied to other first order plus dead time processes, the tuning constants have been normalized as follows. Proportional gain kc is reported as the product KC·KL where KL = $K_M \cdot K_P \cdot K_V$. In the case of a mixing process, the gain $K_P = 1$. The integral time τ_I is reported as τ_1/τ_I. The integral of the absolute value of the error is reported as the program value divided by the product $K_M \cdot X_B$. When the integral of the absolute value of the error is normalized in this manner it can be thought of as having units equal to the product of the disturbance magnitude xb = 1 and time in units of τ_1.

The range of values over which K_C and τ_I were allowed to vary correspond to the range of values over which a typical commercial PID controller would be expected to extend. The results of these calculations, illustrated in the contour plots, show that the response surface of the integral of the absolute value of the error has a single minimum value in each case. Sood and Huddleston [16] reported in their study that a bimodal response surface occurred under similar conditions. They found two local minimums: one of these near the tuning constants predicted by conventional correlations and a second minimum corresponding to higher values of K_C and τ_I. The presence of valve constraints in the present study may provide an explanation for the difference in results. In the absence of valve constraints an artificial minimum may be found for high values of proportional gain. However, in an actual process high proportional gain results in a saturated valve and no further improvement in control system performance. This conclusion depends upon the magnitude of the load disturbance. There may be cases with extremely small disturbance magnitudes for which a bimodal response surface would result. This condition did not appear to occur in any of the optimization runs performed in this study.

Optimization Program and Results

The program given in Figure 5 was suitable for generating a response surface. However, this program would require too many evaluations of function object if it were intended to find exact values of the tuning constants that give the minimum integral of the absolute value of the error. In order to find optimum tuning constants more efficiently, the procedure of H. H. Rosenbrock [6] was employed. This procedure is capable of finding the minimum of a multivariable, unconstrained, nonlinear function. The procedure is based on a direct search method and does not require calculation of derivatives. The version of this procedure employed in this study was adapted from the FORTRAN source code developed by A. I. Johnson [6]. The FORTRAN code was converted to Pascal and function object from Figure 5 was inserted for use as the objective function. The program was compiled by the Pascal 8000 Version 2.0 compiler available on the University Computer Center's IBM 3081D mainframe computer. The adapted program listing is given in Appendix A. The Pascal 8000 Version 2.0 compiler generates machine code that performs floating point calculations in IBM double precision format.

The results of the optimum tuning constant calculations are given in Figures 13 to 16 for the proportional-integral controller and in

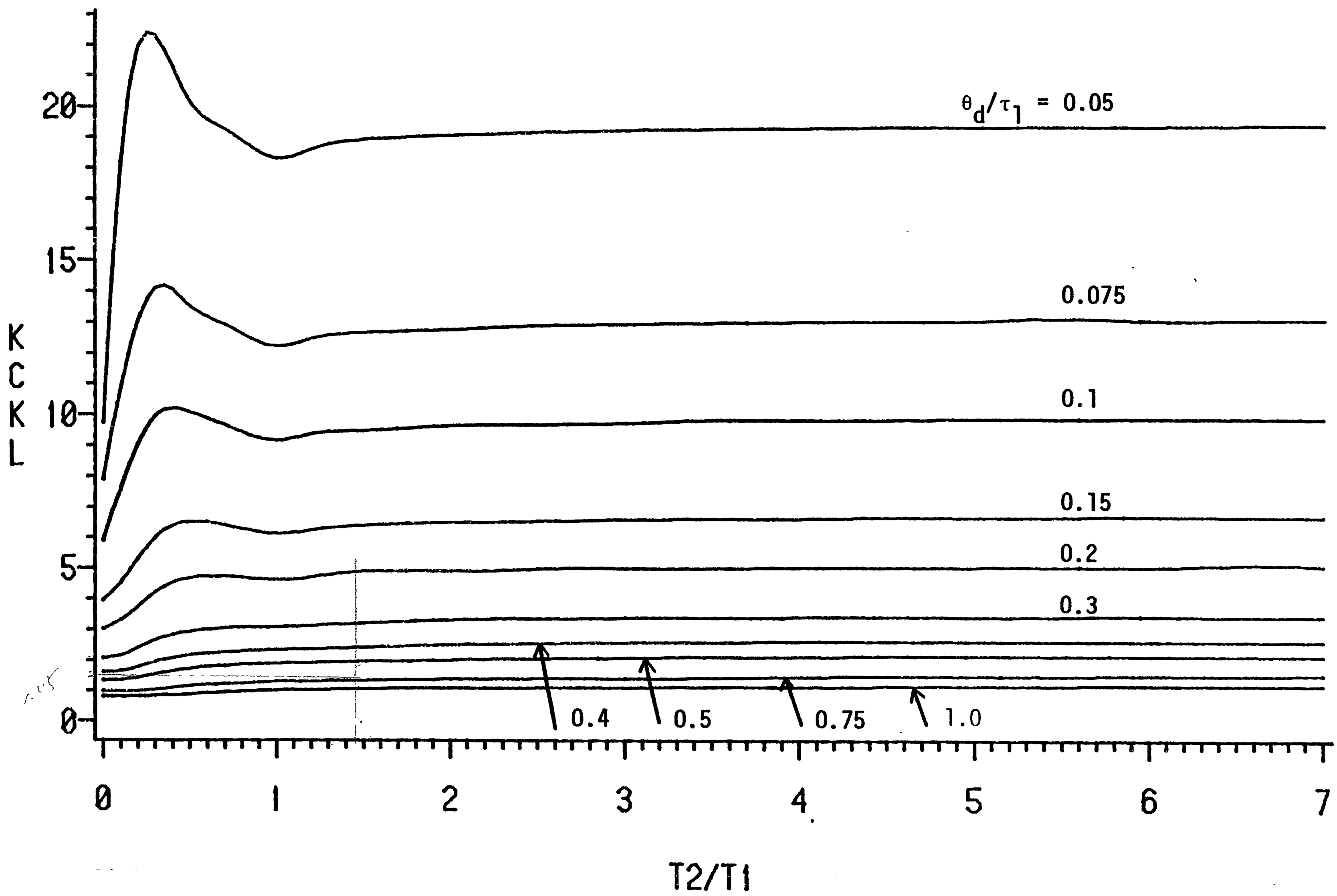


Figure 13. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PI Controller Proportional Gain

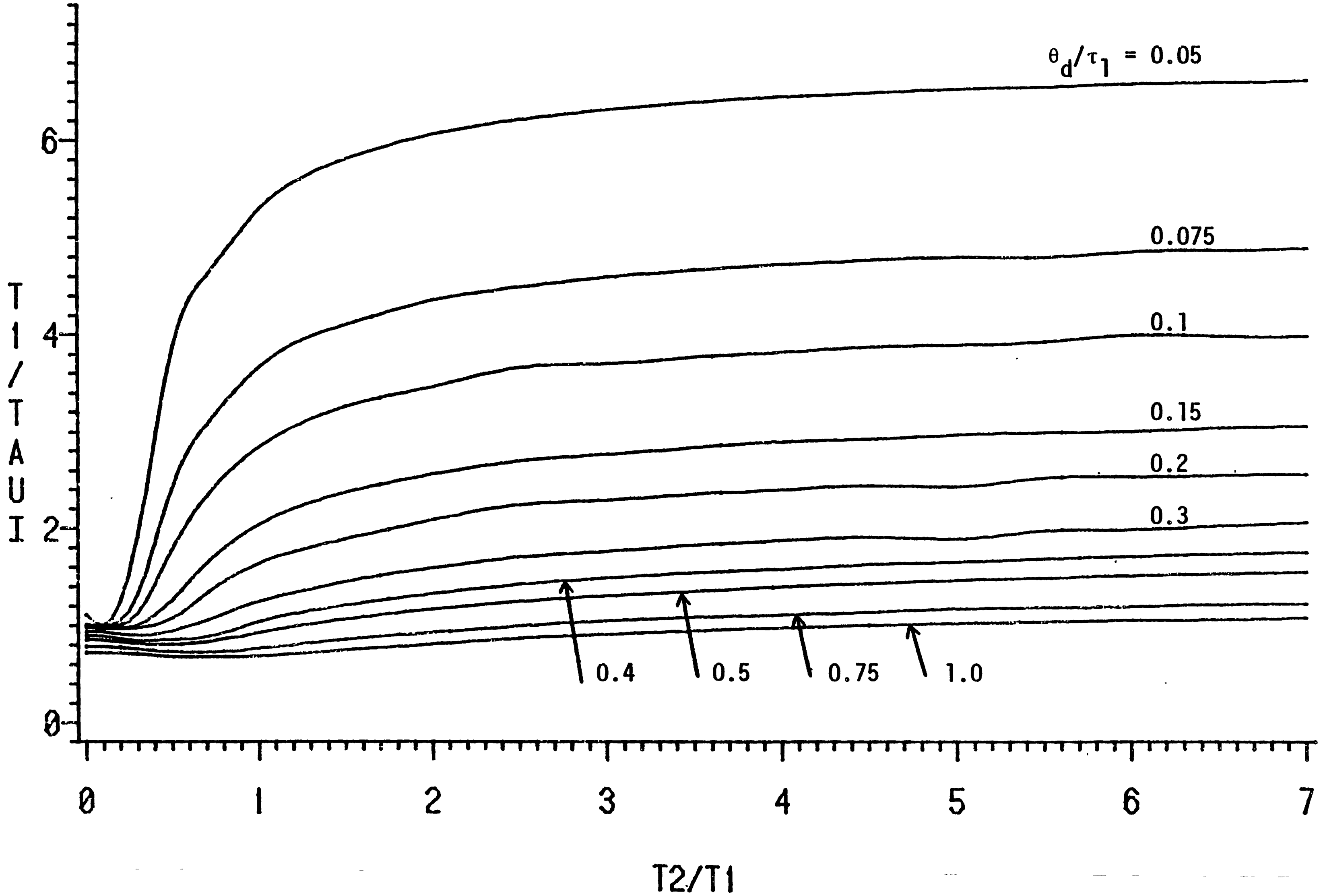


Figure 14. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PI Controller Integral Time

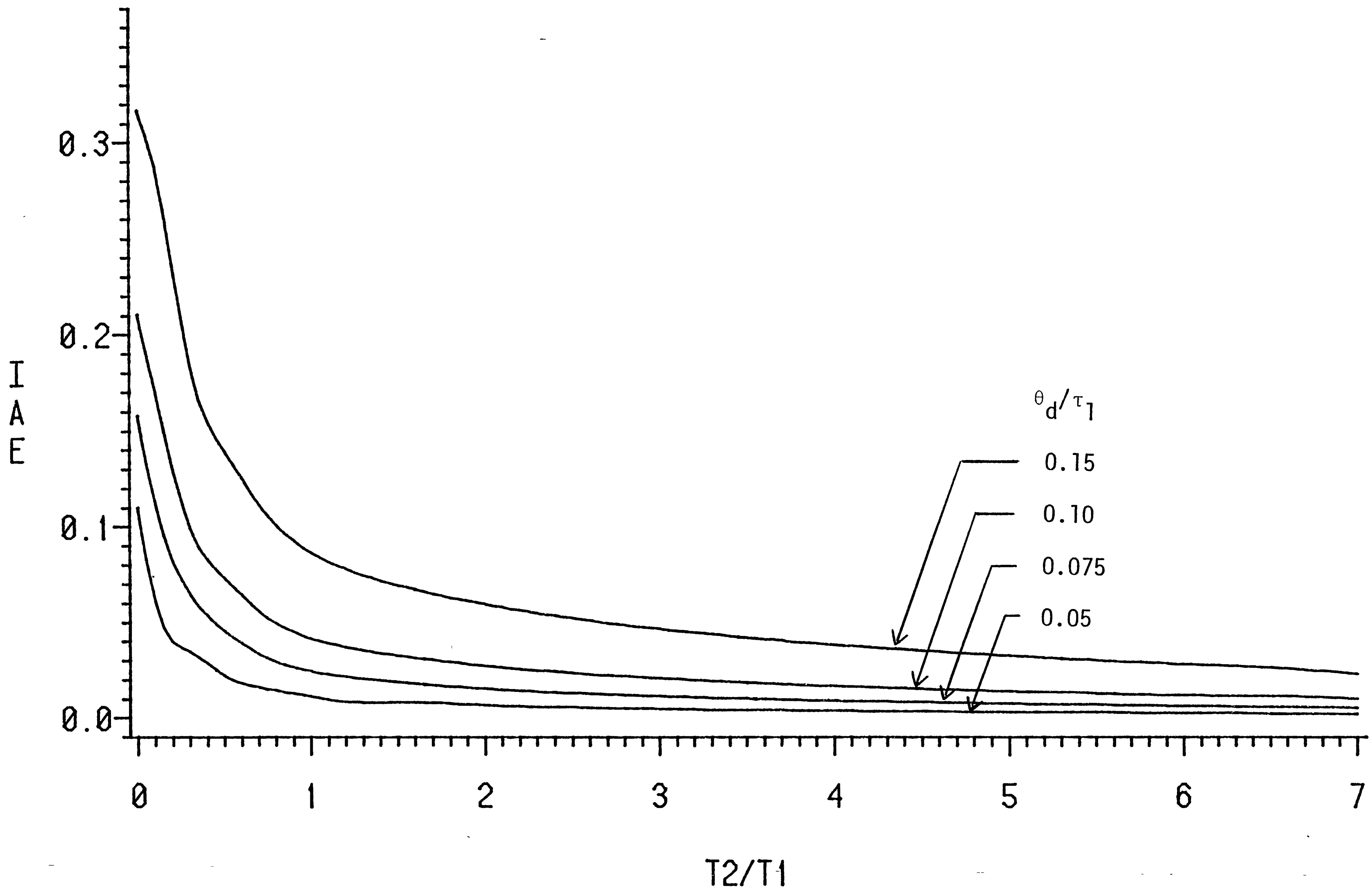


Figure 15. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PI Controller IAE, Low Range of θ_d/τ_1

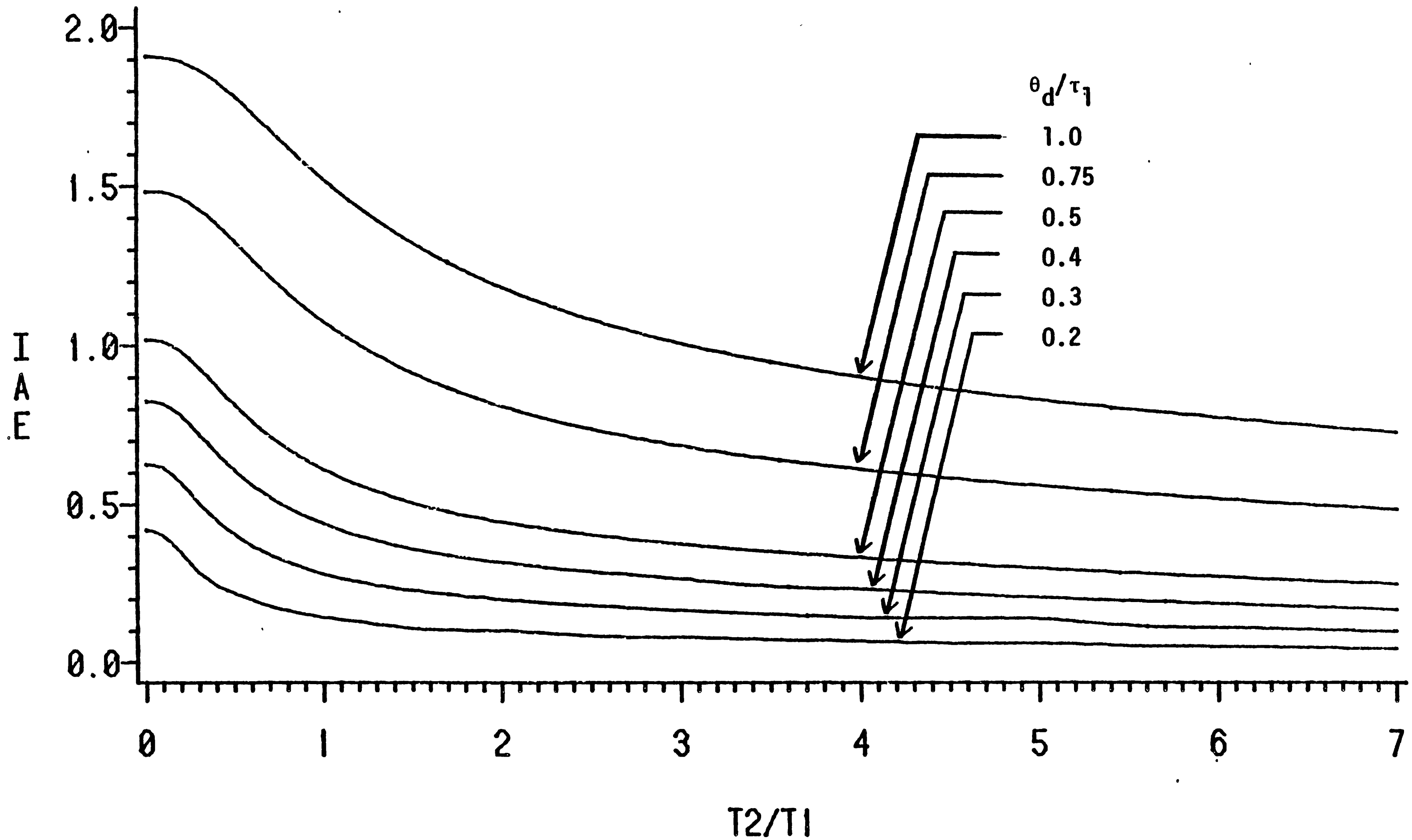


Figure 16. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PI Controller IAE, High Range of θ_d/τ_1

Figures 17 to 21 for the proportional-integral-derivative controller. In these figures the normalized values $K_C \cdot K_L$, τ_1/τ_I, and IAE are given as a function of the ratios θ_d/τ_1 and τ_2/τ_1. The figures were prepared by entering the results of the optimization program into a plotting package available from SAS Institute [14].

In each of these plots the ratio τ_2/τ_1 is given as the abscissa. The normalized tuning constant or IAE value is indicated as the ordinate. A separate line is drawn for each value of the parameter θ_d/τ_1. In this study the simulation time tt was set equal to six times the value of the ultimate period as calculated from frequency response considerations. The magnitude of the step change in disturbance variable xb was set equal to 100 mg/L for the PI controller runs and to 10 mg/L for the PID controller runs.

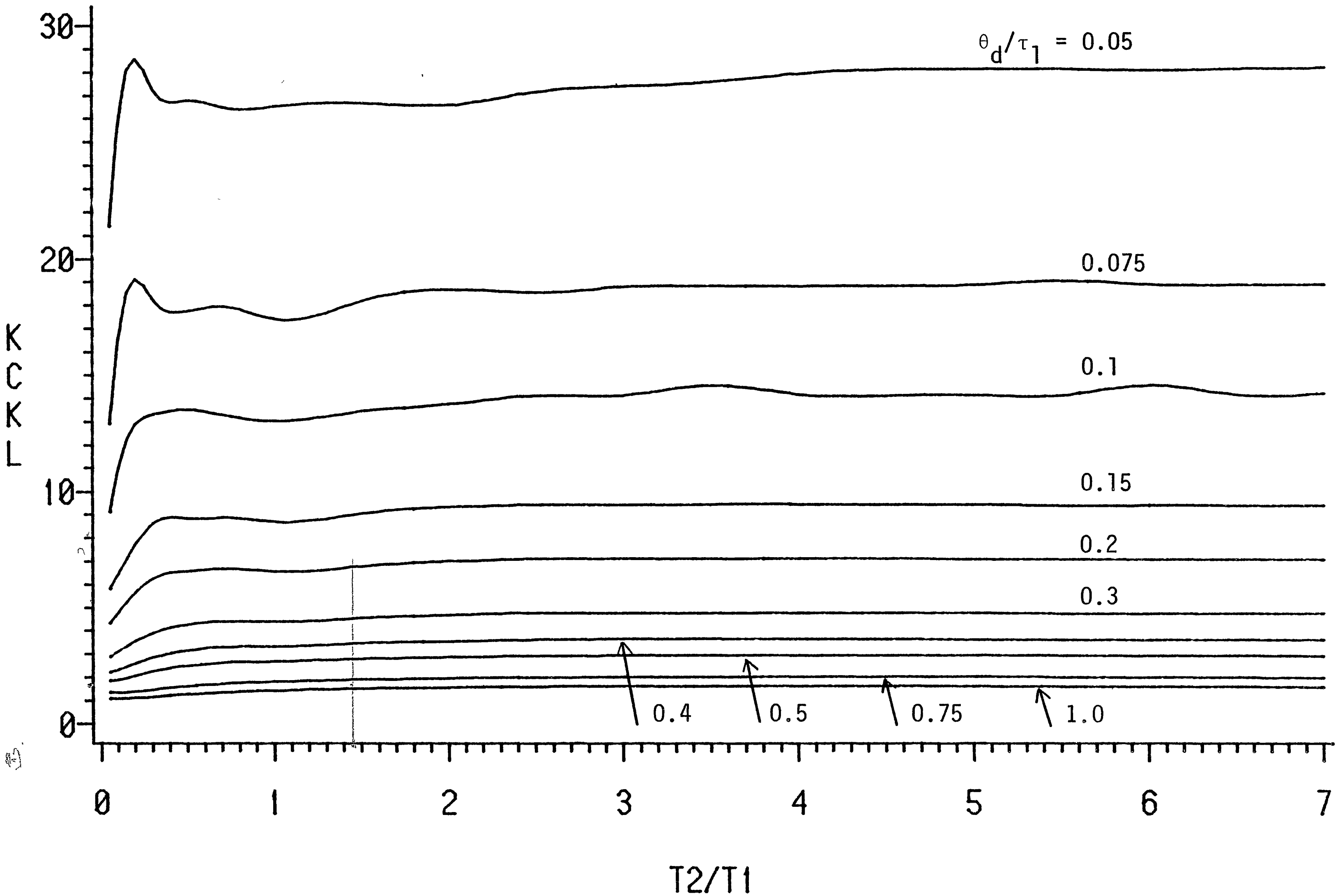


Figure 17. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PID Controller Proportional Gain

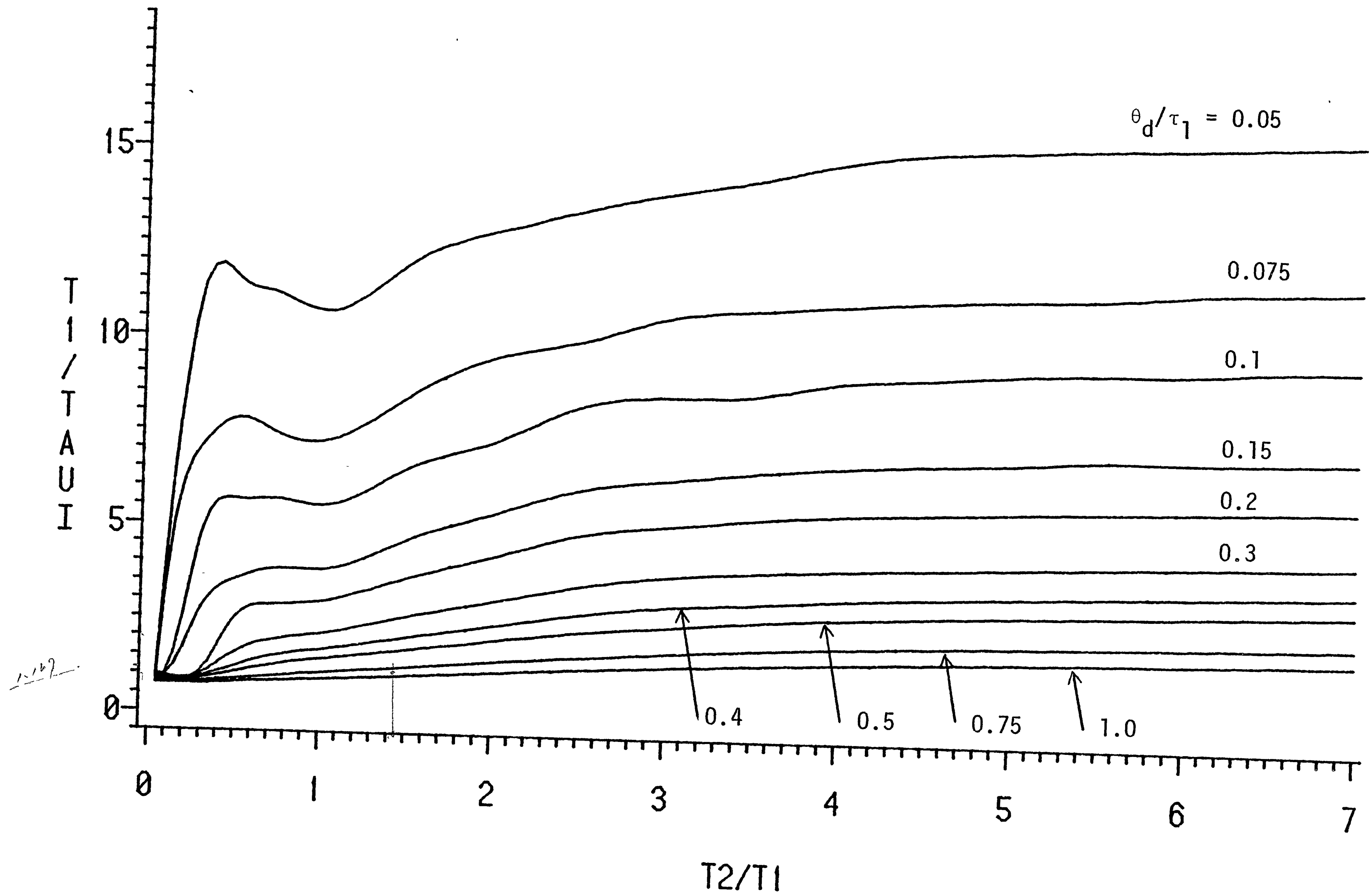


Figure 18. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PID Controller Integral Time

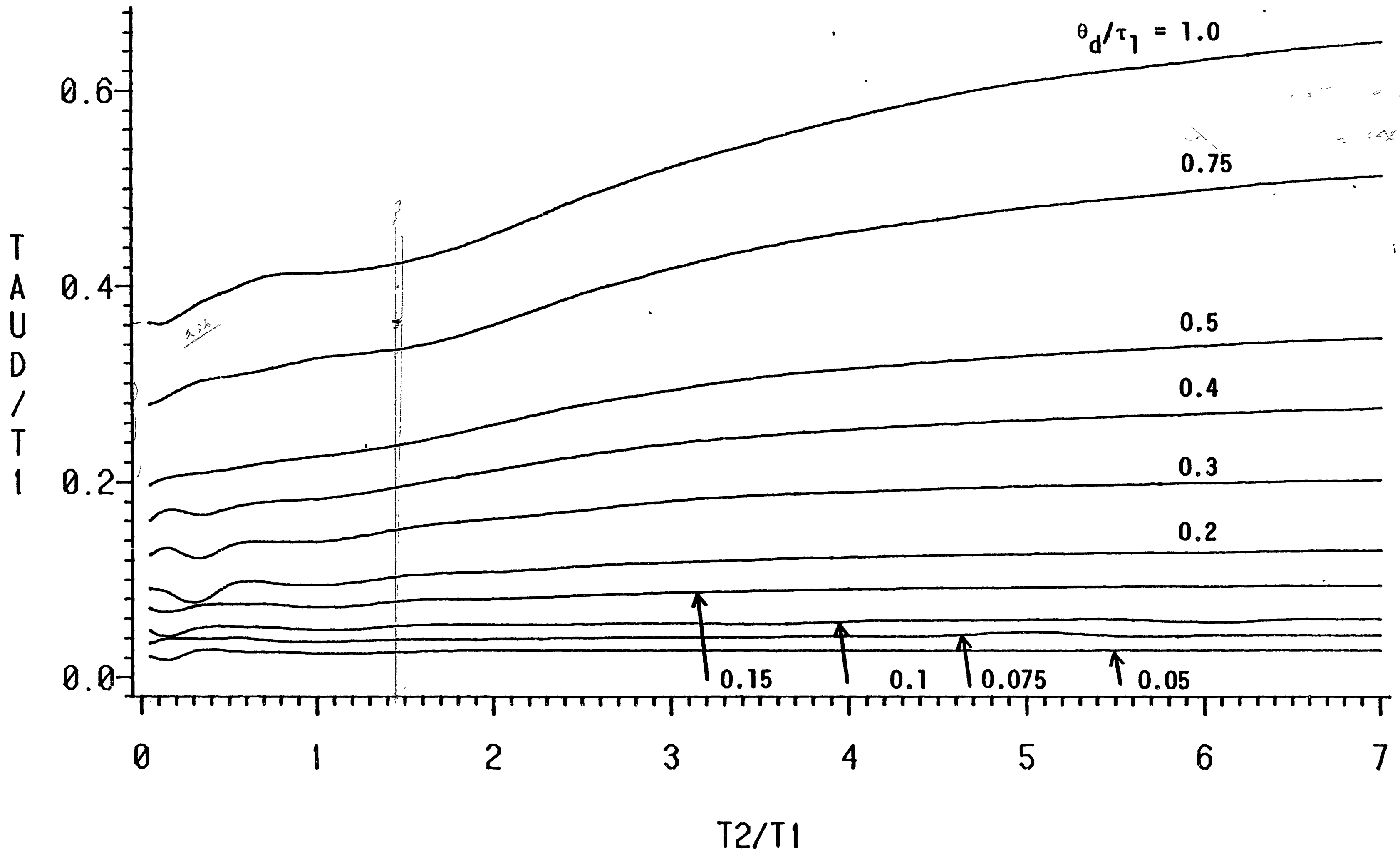


Figure 19. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PID Controller Derivative Time

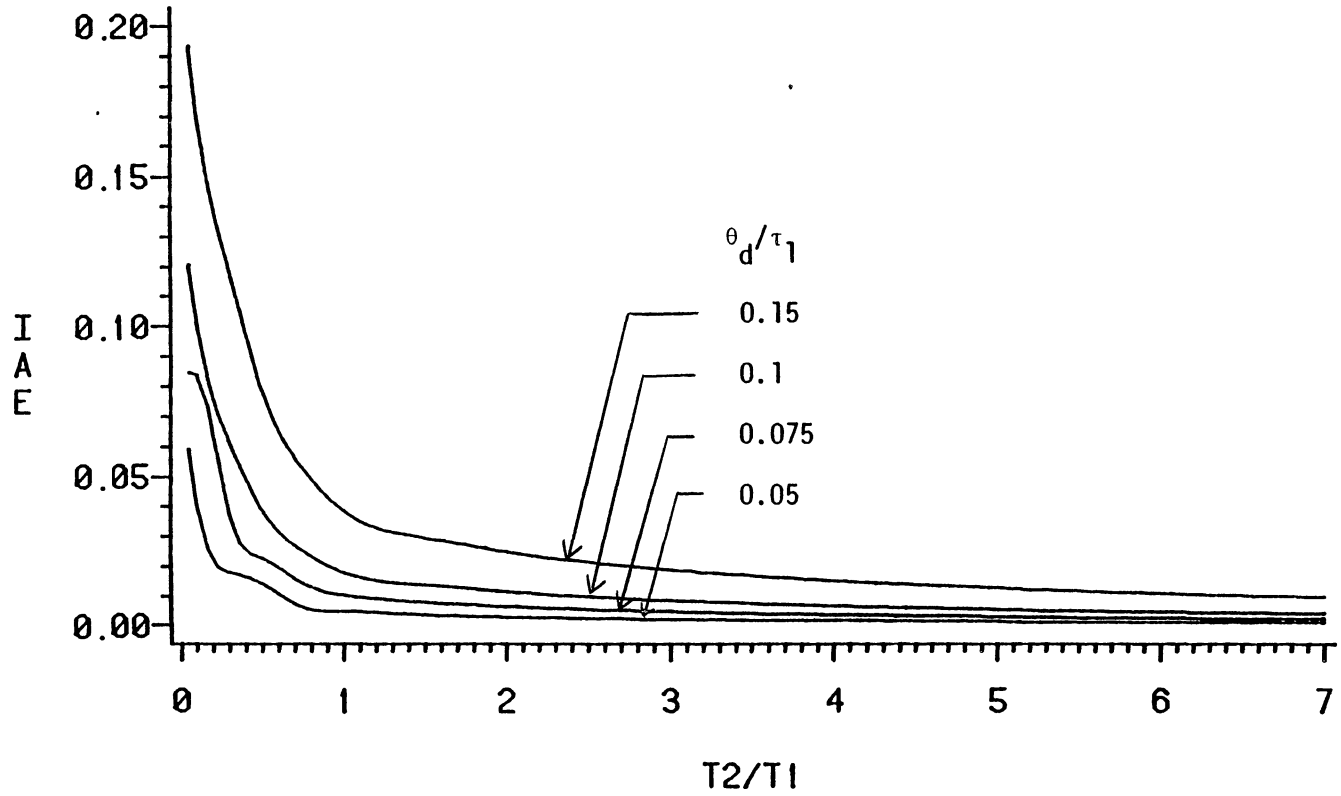


Figure 20. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PID Controller IAE, Low Range of θ_d/τ_1

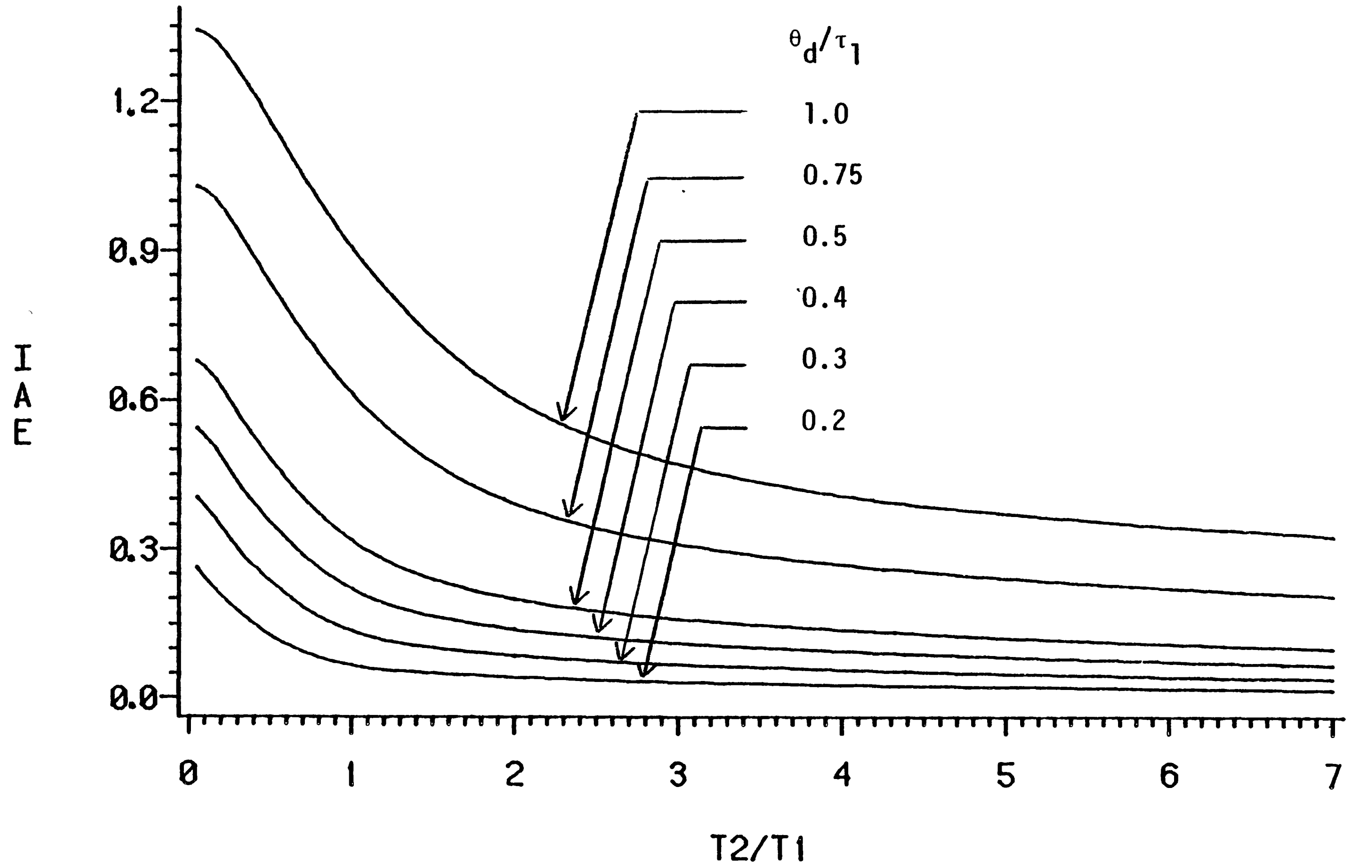


Figure 21. Effect of θ_d/τ_1 and τ_2/τ_1 on Optimum PID Controller IAE, High Range of θ_d/τ_1

CHAPTER IV

CORRELATION OF RESULTS

The usefulness of the data presented in graphical form in Chapter III may be improved by reducing this data to the form of mathematical equations. This was accomplished by finding a correlation to fit each of the curves presented in Figures 13 to 21. In the case of the PI controller each curve was based on optimization runs for 19 different values of τ_2/τ_1. In the case of the PID controller each curve was based on optimization runs for 18 different values of τ_2/τ_1. A total of ten curves, corresponding to ten different values of the ratio θ_d/τ_1, were generated for each tuning constant or IAE function. In the majority of cases the form of these curves suggested that a nonlinear function would provide the best fit.

The method used to fit candidate functions to the available data was based on a least squares approach. The basic idea may be explained as follows. A particular curve with n data points will have n values of the independent variable x_1, x_2, ..., x_n and n values of the dependent variable y_1, y_2, ..., y_n. The nonlinear function to be fitted to the data will predict values of the dependent variable y_1', y_2', ..., y_n'. The sum $(y_1-y_1')^2 + (y_2-y_2')^2 + \cdots + (y_n-y_n')$ is called the sum of the squared residuals. The lower this sum is, the better the curve fits. In the case of a multiparameter nonlinear function an optimization procedure is required to find the function parameters that give the minimum value of the sum of the squared residuals.

The optimization program used for curve fitting in this study was based on the simplex procedure described by Hedler and Mead [10]. The program source code was adapted from the Pascal code developed by Caceci and Cacheris [2]. Minor changes were made in the code to allow compilation under Pascal 8000 Version 2.0 and execution on an IBM 3081D mainframe computer. A listing of the adapted source code is given in Appendix B.

The results of the curve fitting runs are given in Tables I to III for the PI controller and in Tables IV to VII for the PID controller. Each table corresponds to a graph given in Chapter III. The form of the equation is given at the top of each table. Parameter values for each θ_d/τ_1 curve are listed below the equation. The equations used for curve fitting varied in the number of parameters. From three to seven parameters were used. The standard deviation of the experimental points from the fitted function is listed with the parameter values for each curve.

Graphs similar to those given in Chapter III were prepared using the fitted equations. These are presented in Figures 22 to 25 for the PI controller and in Figures 26 to 30 for the PID controller. The graphs prepared using the fitted equations may be compared to those based on the optimization runs given in Chapter III.

TABLE I

PI CONTROLLER PROPORTIONAL GAIN CURVE FITTING RESULTS

$$K_C K_L = A - e^{-B \tau_2/\tau_1} [C \cos (D \tau_2/\tau_1) + E \sin (D \tau_2/\tau_1)]$$

θ_d/τ_1	A	B	C	D	E	Standard Deviation
0.05	19.16021053	4.72844283	9.41472472	5.54953579	- 11.85902112	0.26425100
0.075	12.88616553	3.57033096	4.99739025	6.10645031	- 2.12961668	0.21357490
0.1	9.69937357	4.27126621	3.78274534	6.43393407	0.40520229	0.18195879
0.15	6.53568186	10.41222288	2.58537289	0.12389152	200.85546451	0.15181330
0.2	4.97491114	3.97668589	1.95986468	0.01818878	375.60953034	0.09969763
0.3	3.37753881	2.33294842	1.31122357	0.11008651	13.56279186	0.06907022
0.4	2.59853768	1.40585127	1.03177237	$-2.96438521 \times 10^{-5}$	2.80523694	0.03455487
0.5	2.12850671	1.18670059	0.82442081	$-4.99709384 \times 10^{-9}$	0.25032335	0.02964337
0.075	1.49467677	0.84932879	0.54437582	$-8.00972831 \times 10^{-9}$	- 0.11627429	0.02740568
1.0	1.17141091	0.79911293	0.39684857	$-2.96914688 \times 10^{-7}$	0.85928797	0.01649091

TABLE II

PI CONTROLLER INTEGRAL TIME CURVE FITTING RESULTS

$$\tau_1/\tau_I = A + B/[1 + C\,|\tau_2/\tau_1 - D|^E]$$

θ_d/τ_1	A	B	C	D	E	Standard Deviation
0.05	1.03575855	5.56378574	0.23900248	0.11624811	-1.68145670	0.11903690
0.075	0.95977007	3.95561919	0.37403148	0.14522281	-1.55864968	0.07274715
0.1	0.89105385	3.16309931	0.49717305	0.16670167	-1.48052852	0.04187496
0.15	0.87341458	2.27816923	0.77294626	0.18831337	-1.41494759	0.03292929
0.2	0.89891052	1.75795923	1.13896285	0.17888151	-1.51150947	0.02743033
0.3	0.85369874	1.36709123	1.83667626	0.23273961	-1.29730357	0.02123175
0.4	0.81281238	1.08313962	2.43167287	0.27750494	-1.44756168	0.01513327
0.5	0.78487362	0.90527347	3.18765764	0.26952873	-1.51941578	0.00830626
0.75	0.71993673	0.62152179	3.25032854	0.54881605	-1.46916934	0.00583965
1.0	0.67109941	0.51308283	4.16611085	0.64499365	-1.55784624	0.00434519

TABLE III

PI CONTROLLER IAE CURVE FITTING RESULTS

$$IAE = A + B/(C + \tau_2/\tau_1)$$

θ_d/τ_1	A	B	C	Standard Deviation
0.05	0.00016723	0.01378123	0.12549148	0.00077133
0.075	0.00064771	0.03220387	0.20193263	0.00185386
0.1	0.00234711	0.05565353	0.25966571	0.00393629
0.15	0.00896566	0.11697854	0.36410559	0.00764922
0.2	0.02191948	0.19405771	0.45977776	0.01055301
0.3	0.04879802	0.41847755	0.68491163	0.01510704
0.4	0.07446595	0.74031452	0.93355260	0.01860096
0.5	0.10491451	1.13974012	1.18386141	0.02256713
0.75	0.12132195	2.60949681	1.88552414	0.03216572
1.0	0.21403862	4.64313221	2.60662201	0.03915691

TABLE IV

PID CONTROLLER PROPORTIONAL GAIN CURVE FITTING RESULTS

$$K_C K_L = A - e^{-B\, \tau_2/\tau_1} [C \cos (D\, \tau_2/\tau_1) + E \sin (D\, \tau_2/\tau_1)]$$

θ_d/τ_1	A	B	C	D	E	Standard Deviation
0.05	27.56693666	3.21738804	-2.33363278	-31.41592903	-7.26624771	0.61257669
0.075	18.81722956	1.27136574	-1.88230685	-31.41594697	-6.29901806	0.27237931
0.1	13.98379931	7.61663918	6.88964879	$-1.90701633 \times 10^{-6}$	-9.92116294	0.39639633
0.15	9.33177269	4.92329424	4.52672239	$-3.40030676 \times 10^{-7}$	-0.09780655	0.20657473
0.2	7.02101441	3.98644204	3.25368254	$-1.04632894 \times 10^{-5}$	1.92277126	0.19590525
0.3	4.71104824	2.93827302	2.09468446	$-5.58866142 \times 10^{-8}$	1.65373757	0.10282611
0.4	3.61555083	2.26432807	1.55456659	$-9.04129146 \times 10^{-9}$	2.31434075	0.04285323
0.5	2.93976822	1.89746059	1.21075675	$-9.45291672 \times 10^{-9}$	1.58608696	0.02754515
0.75	2.01637991	2.02899764	0.75478338	0.42772479	1.58452412	0.01448776
1.0	1.58047933	0.61539767	0.53453858	0.36030766	-0.67787164	0.00745211

TABLE V

PID CONTROLLER INTEGRAL TIME CURVE FITTING RESULTS

$$\tau_1/\tau_I = A + B\,(1 - e^{C\,\tau_2/\tau_1}) + D\,e^{E\,\tau_2/\tau_1}\,\sin\,(F\,\tau_2/\tau_1 + G)$$

θ_d/τ_1	A	B	C	D	E	F	G	Standard Deviation
0.05	7.88907452	8.14914425	-0.47062819	14.01989568	-3.69869605	4.64045952	-0.89862912	0.12717321
0.075	3.75329511	8.08933281	-0.58796694	11.35325141	-3.45143952	3.50785419	-0.48184674	0.11716931
0.1	1.16760318	8.57513521	-0.64589925	5.23824737	-1.70791523	3.17936407	-0.29467022	0.10279623
0.15	0.96313324	6.39963428	-0.61037063	1.70977301	-1.34658223	3.51948901	-0.45078413	0.09277195
0.2	1.19892513	4.96868301	-0.49551792	0.63954896	-1.30803185	7.52194596	-2.72127032	0.07978999
0.3	0.54029512	4.04983024	-0.51956251	0.14045051	-0.23661262	1.80896011	-4.49313921	0.07447438
0.4	2.50699154	7.88550725	-0.02328223	1.79414506	-0.51714053	0.55372792	-1.81570416	0.03919181
0.5	2.63813309	0.73850854	-0.28088387	3.56103232	-0.55089769	0.19516987	-2.58373021	0.01953243
0.75	0.64002991	2.51283455	-0.17732943	0.08251026	-0.01000000	1.37725113	-4.72052031	0.00905795
1.0	1.18214772	2.65217723	-0.04453823	0.65282817	-0.49166843	0.56746896	-2.36110017	0.00574148

TABLE VI

PID CONTROLLER DERIVATIVE TIME CURVE FITTING RESULTS

$$\tau_D/\tau_1 = A + B\, e^{C\, \tau_2/\tau_1}$$

θ_d/τ_1	A	B	C	Standard Deviation
0.05	$2.69744291 \times 10^{-2}$	$-9.04199529 \times 10^{-3}$	-4.17585174	0.00137847
0.075	$-1.49103557 \times 10^{3}$	1.49107308×10^{3}	$6.22270036 \times 10^{-7}$	0.00122542
0.1	$-4.82923038 \times 10^{3}$	4.82927885×10^{3}	$3.81252783 \times 10^{-7}$	0.00231682
0.15	$-1.22459823 \times 10^{3}$	1.22466996×10^{3}	$3.16420439 \times 10^{-6}$	0.00289871
0.2	$-1.36347559 \times 10^{4}$	1.36348467×10^{4}	$5.01643551 \times 10^{-7}$	0.00590471
0.3	$-3.11294460 \times 10^{4}$	3.11295781×10^{4}	$3.90109862 \times 10^{-7}$	0.00832926
0.4	$-3.15391753 \times 10^{4}$	3.15394574×10^{4}	$5.62369948 \times 10^{-7}$	0.00960051
0.5	$-4.09010504 \times 10^{4}$	4.09012582×10^{4}	$5.66236090 \times 10^{-7}$	0.01124117
0.75	$-5.24077933 \times 10^{4}$	5.24080856×10^{4}	$6.82590779 \times 10^{-7}$	0.01387142
1.0	$-5.93244716 \times 10^{4}$	5.93248430×10^{4}	$7.52191641 \times 10^{-7}$	0.01507123

TABLE VII

PID CONTROLLER IAE CURVE FITTING RESULTS

$$IAE = A + B/(C + \tau_2/\tau_1)$$

θ_d/τ_1	A	B	C	Standard Deviation
0.05	-0.00024367	0.00657028	0.06113477	0.00073655
0.075	0.00023459	0.00828941	0.05871175	0.00129804
0.1	-0.00040527	0.02617856	0.16326077	0.00200234
0.15	0.00007832	0.05809119	0.24547782	0.00403015
0.2	0.00205375	0.09754412	0.31399448	0.00527894
0.3	0.00802658	0.20646372	0.45494964	0.00816328
0.4	0.01529003	0.35199421	0.59628164	0.01139624
0.5	0.02187133	0.59332277	0.74289301	0.01508506
0.75	0.03193122	1.24582634	1.14769063	0.02651841
1.0	0.04132849	2.11432235	1.50537644	0.03686955

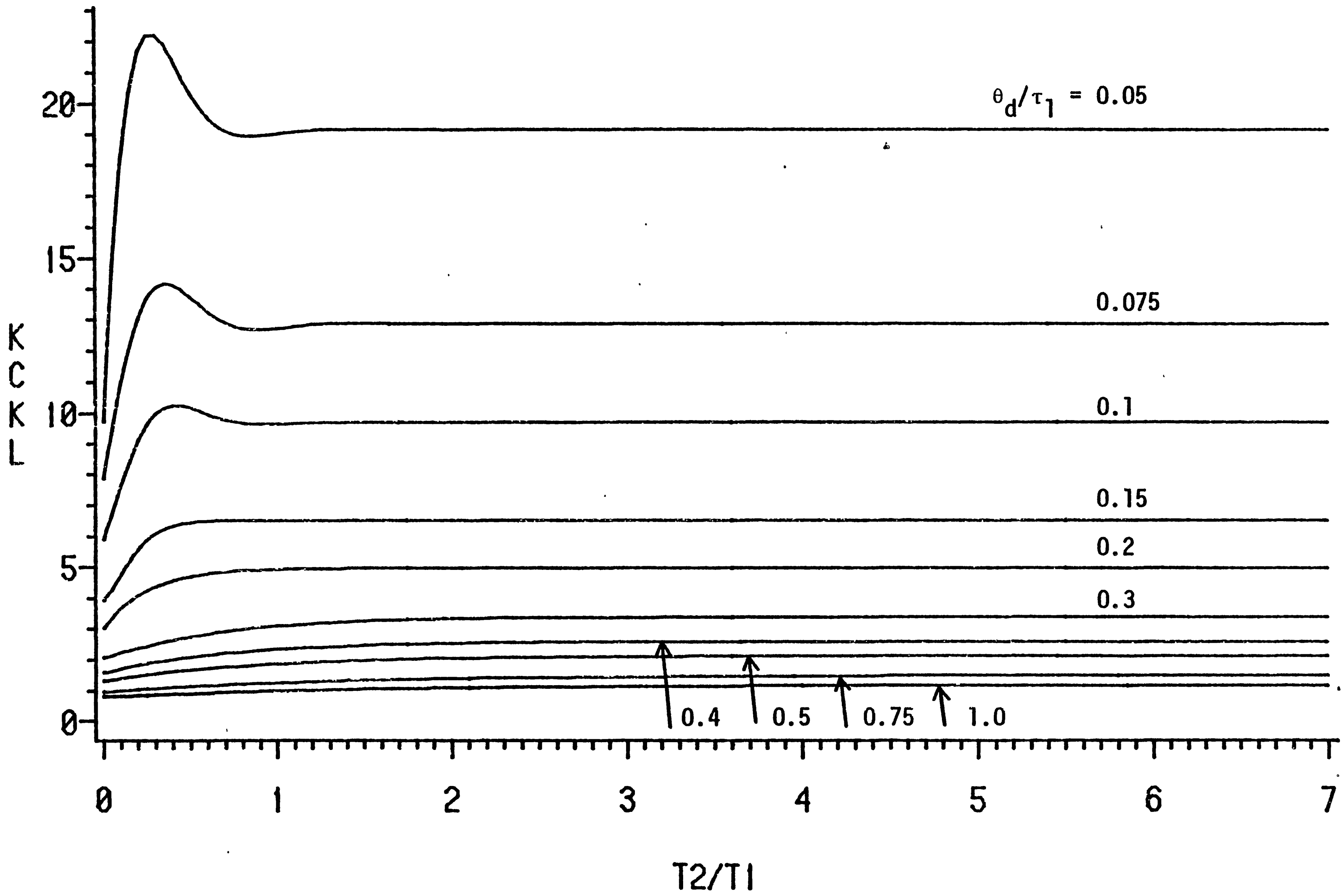


Figure 22. PI Controller Proportional Gain Curve Fitting Results (See Table I)

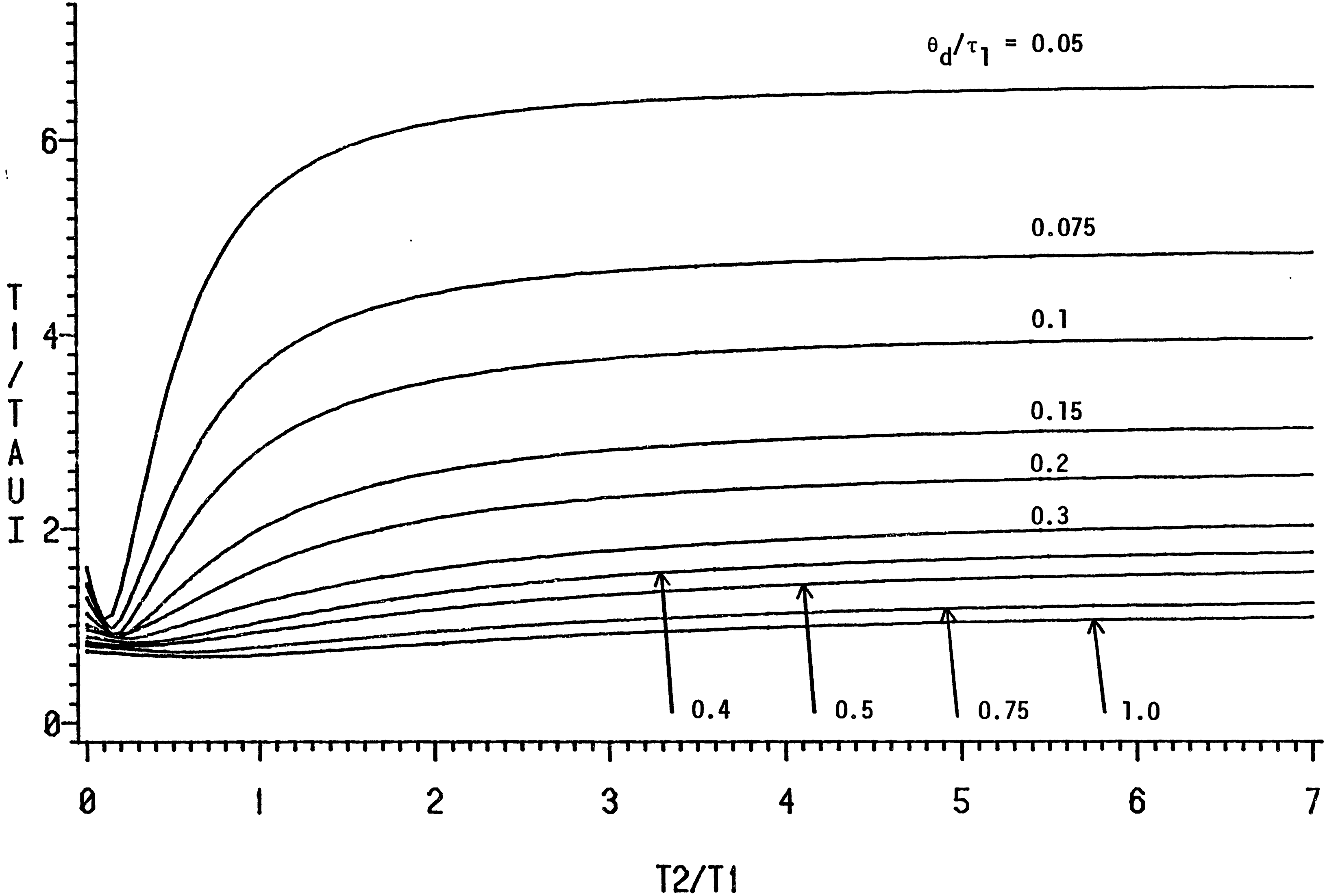


Figure 23. PI Controller Integral Time Curve Fitting Results (See Table II)

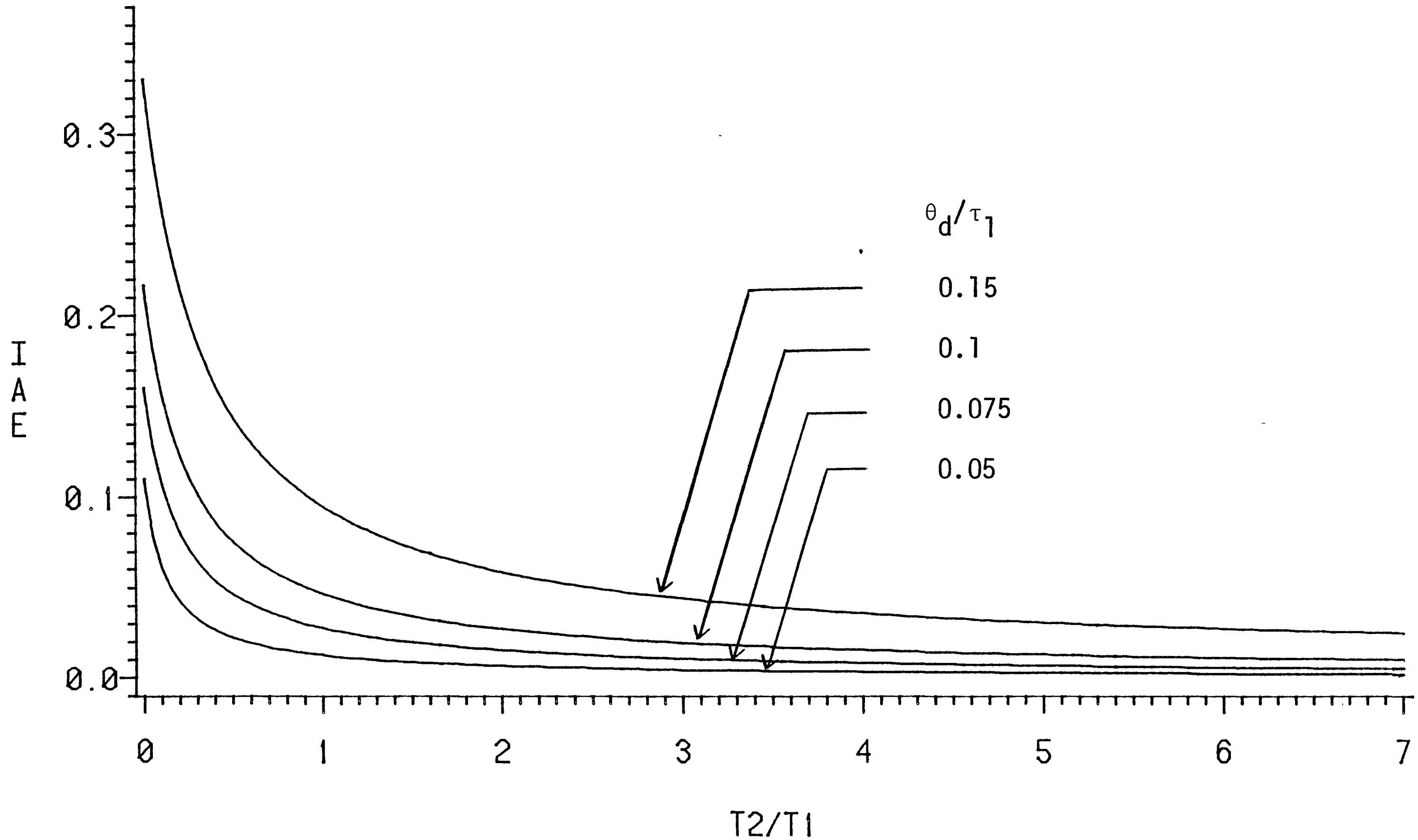


Figure 24. PI Controller IAE Curve Fitting Results, Low Range of θ_d/τ_1 (See Table III)

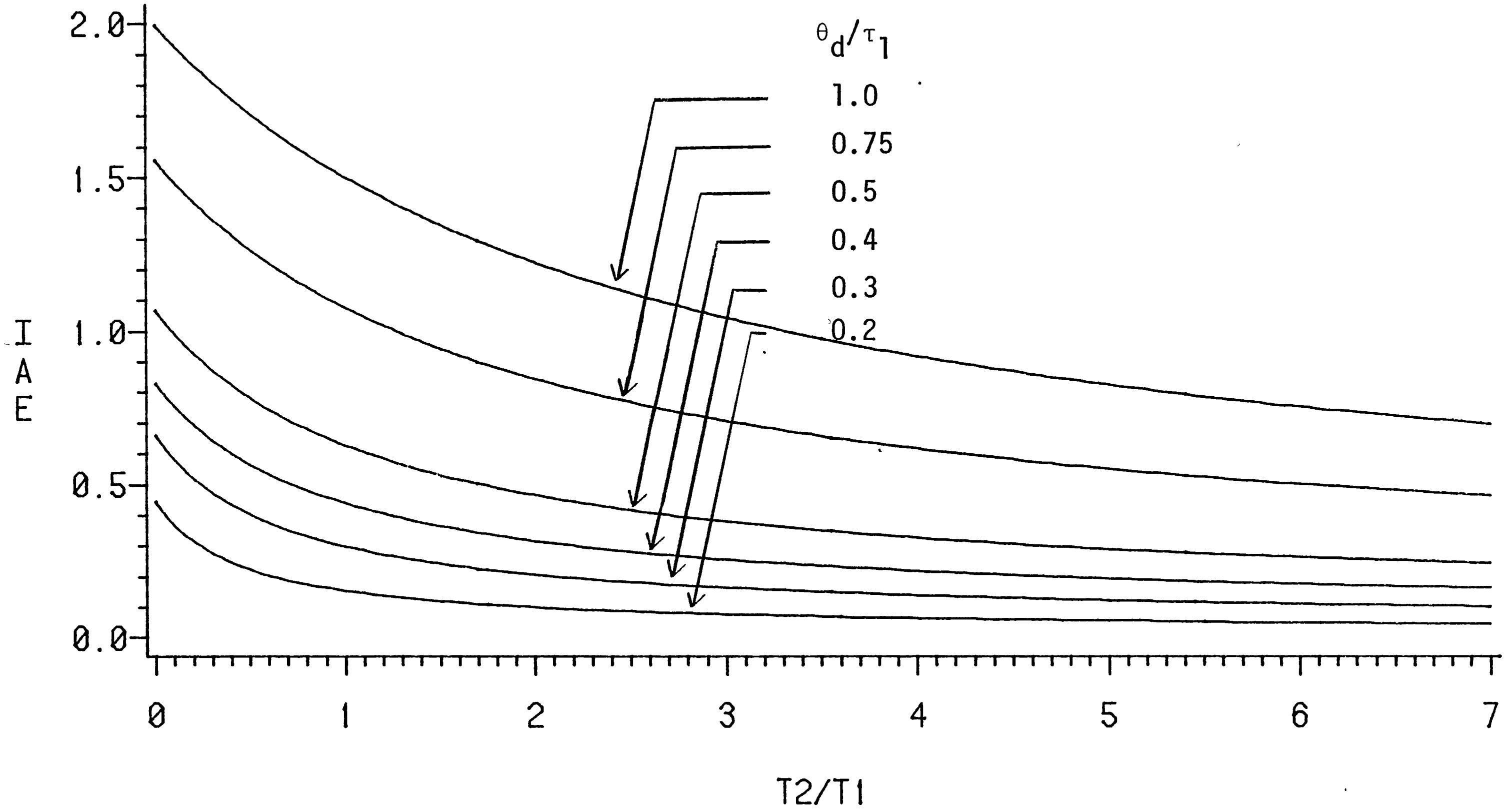


Figure 25. PI Controller IAE Curve Fitting Results, High Range of θ_d/τ_1 (See Table III)

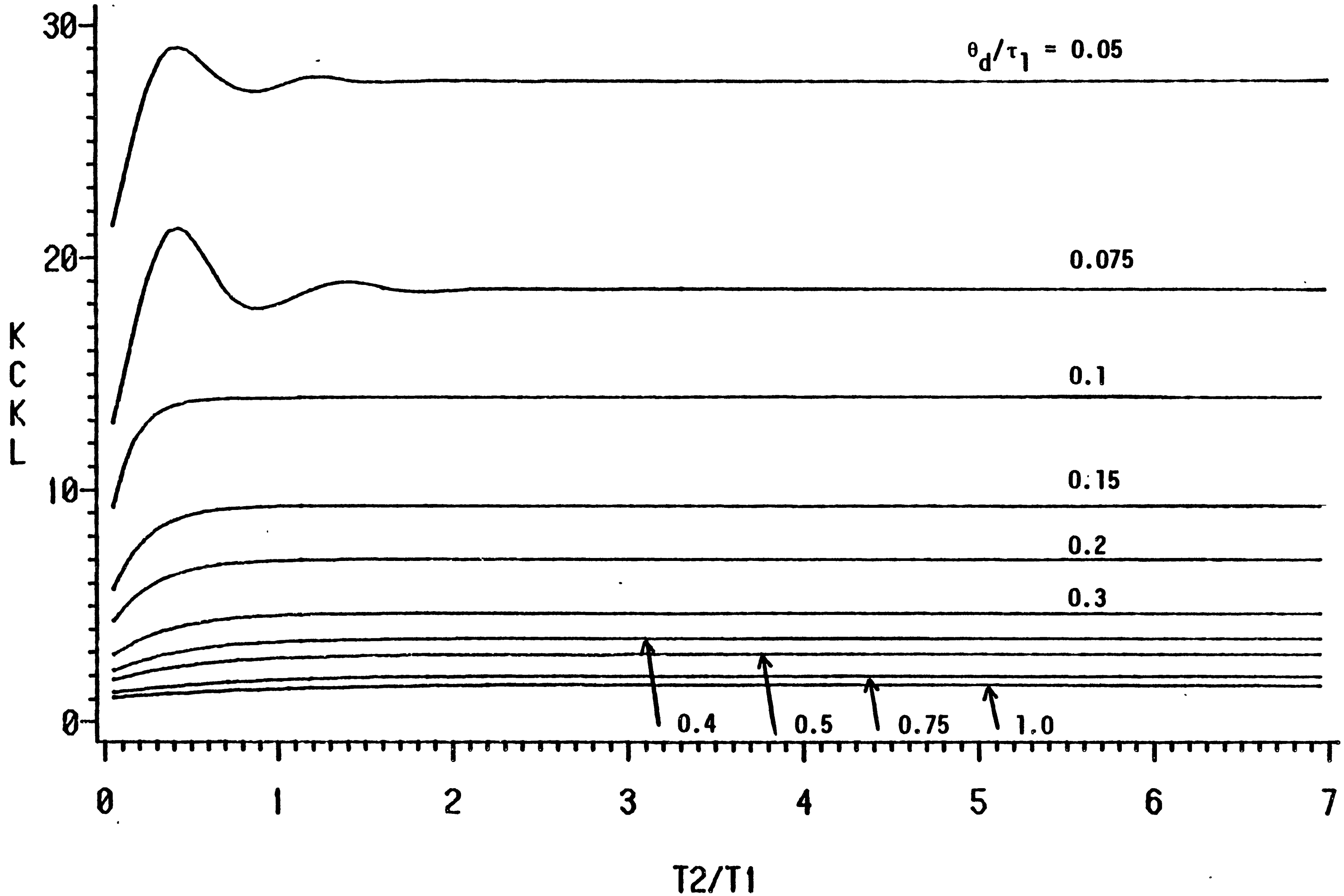


Figure 26. PID Controller Proportional Gain Curve Fitting Results (See Table IV)

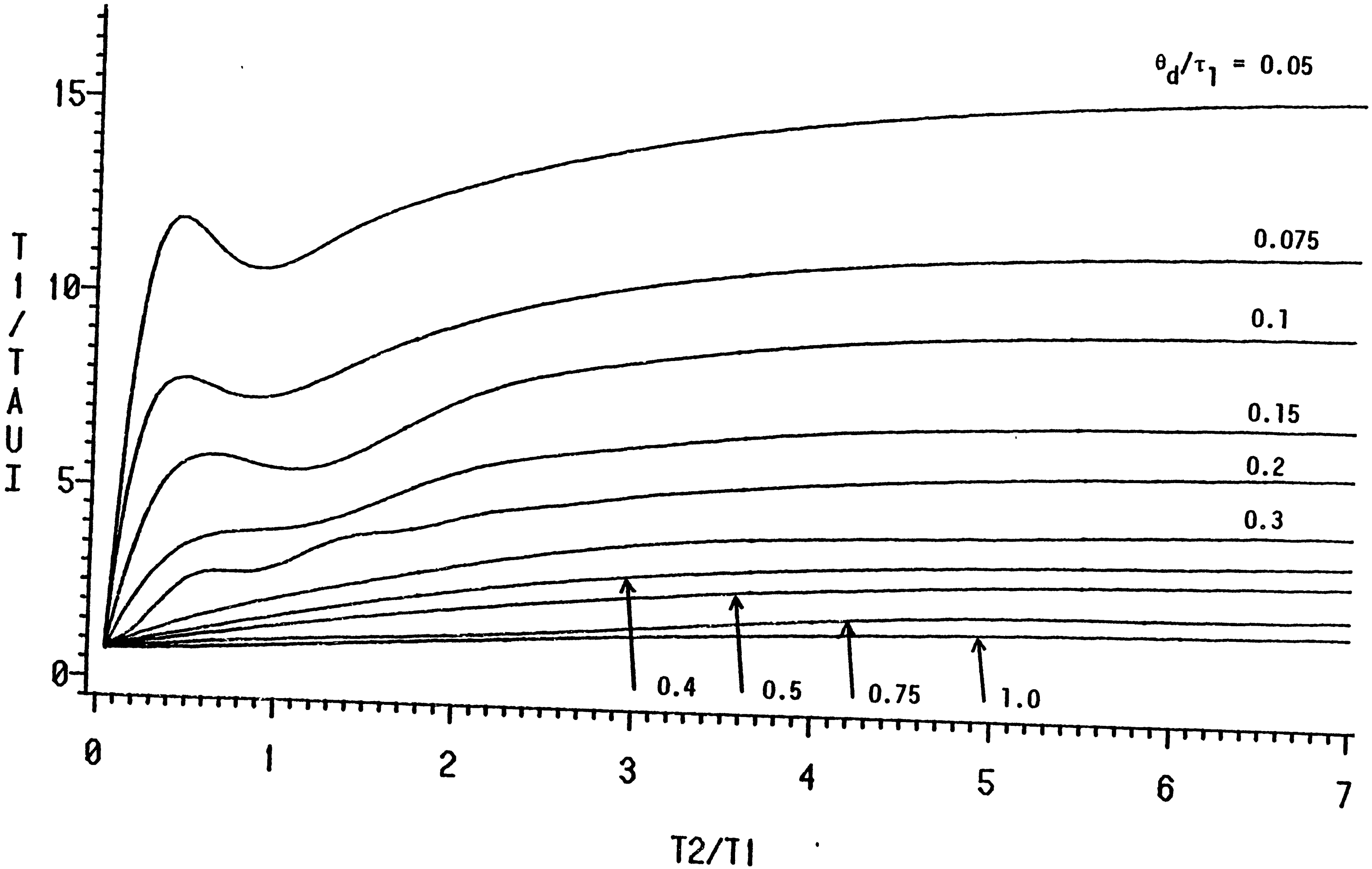


Figure 27. PID Controller Integral Time Curve Fitting Results (See Table V)

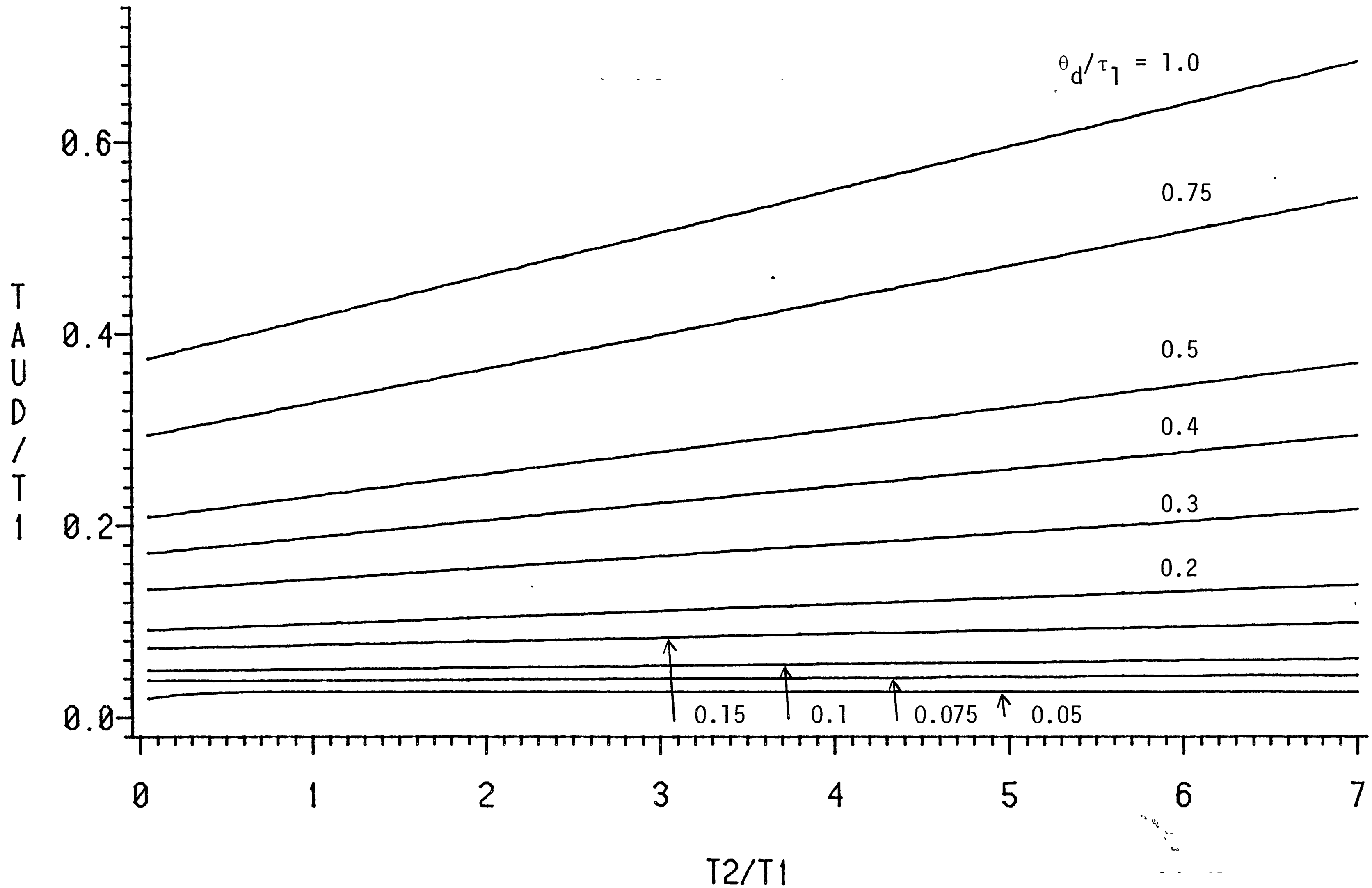


Figure 28. PID Controller Derivative Time Curve Fitting Results (See Table VI)

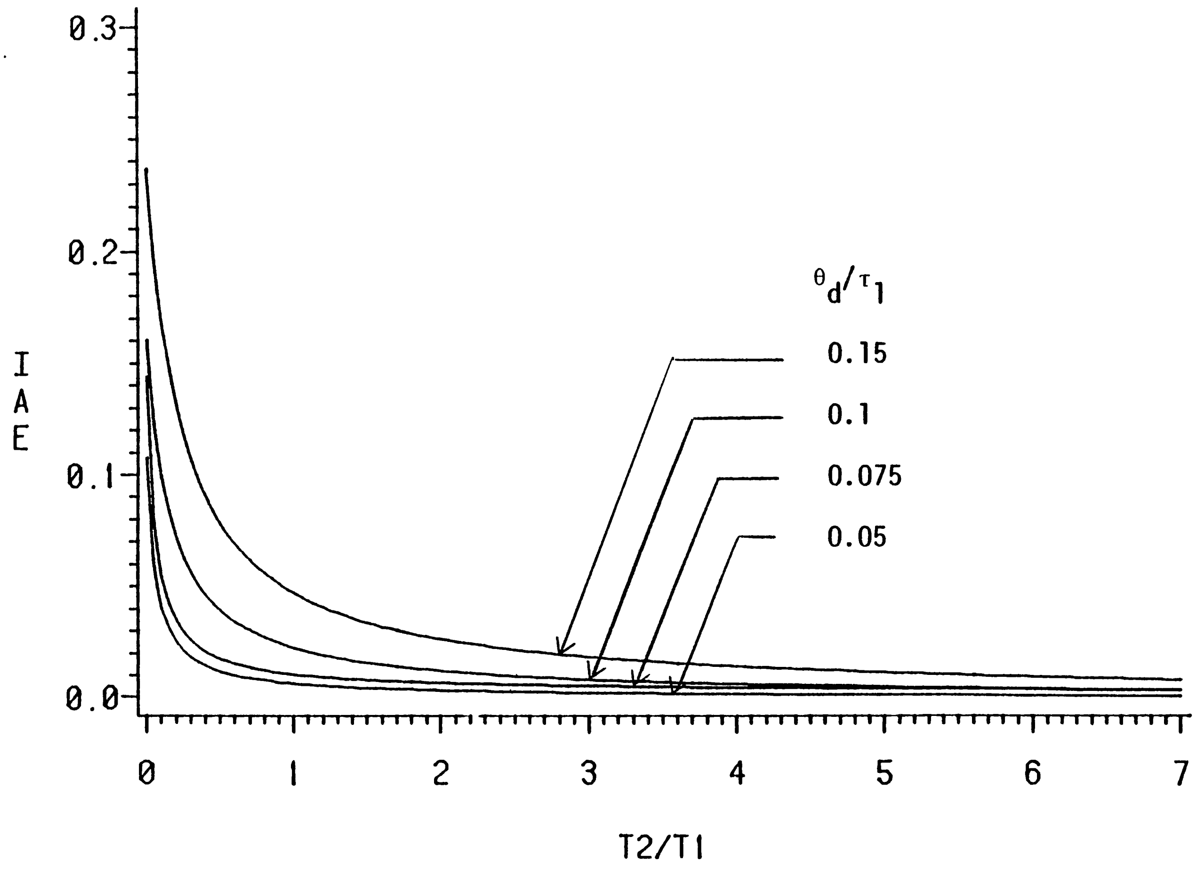


Figure 29. PID Controller IAE Curve Fitting Results, Low Range of θ_d/τ_1 (See Table VII)

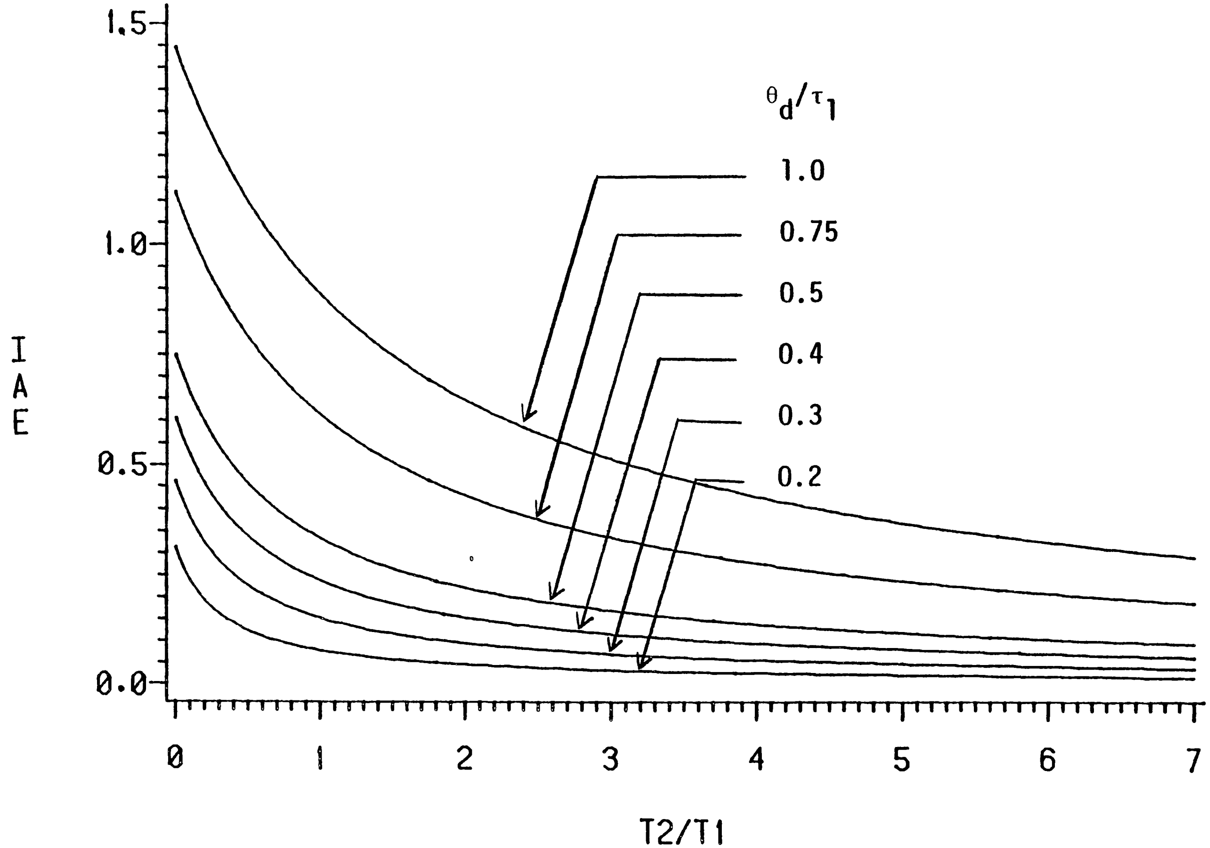


Figure 30. PID Controller IAE Curve Fitting Results, High Range of θ_d/τ_1 (See Table VII)

CHAPTER V

SAMPLE APPLICATIONS

The tuning constants found by the optimization procedure described in Chapter III were used in a number of sample applications in order to illustrate control system performance. Each sample application corresponded to a single calculation of function object given in Figure 5. However, during these runs the program generated a listing of the time response of variables x4, x3, and va. An example of the results from a single sample application are presented in Figures 31 and 32. This sample application uses optimum tuning constants for a PID controller applied to a control system with the following specifications: θ_d/τ_1 = 0.1, τ_2/τ_1 = 0.05, and xb = 10. Figure 31 plots the response of: x4, measurement error after first order lag τ_2; and x3, measured concentration passing the analyzer after imposition of the measurement error. Figure 32 plots the time response of the valve signal va.

The results of the sample application runs are given in Appendix C. In the case of both the PI and PID controllers, runs were performed for θ_d/τ_1 values of 0.1, 0.5, and 1.0. In the case of the PI controller, for each θ_d/τ_1 value the following runs were performed: one setpoint disturbance r = 0.4 (xa = 0, xb = 0); and seven load disturbance runs with xb = 100 and τ_2/τ_1 = 0.1, 0.3, 0.5, 1.0, 2.0, 4.0, and 7.0. In the case of the PID controller, for each θ_d/τ_1 value the following runs were performed: eight load disturbance runs with xb = 10 and τ_2/τ_1 = 0.05, 0.1,

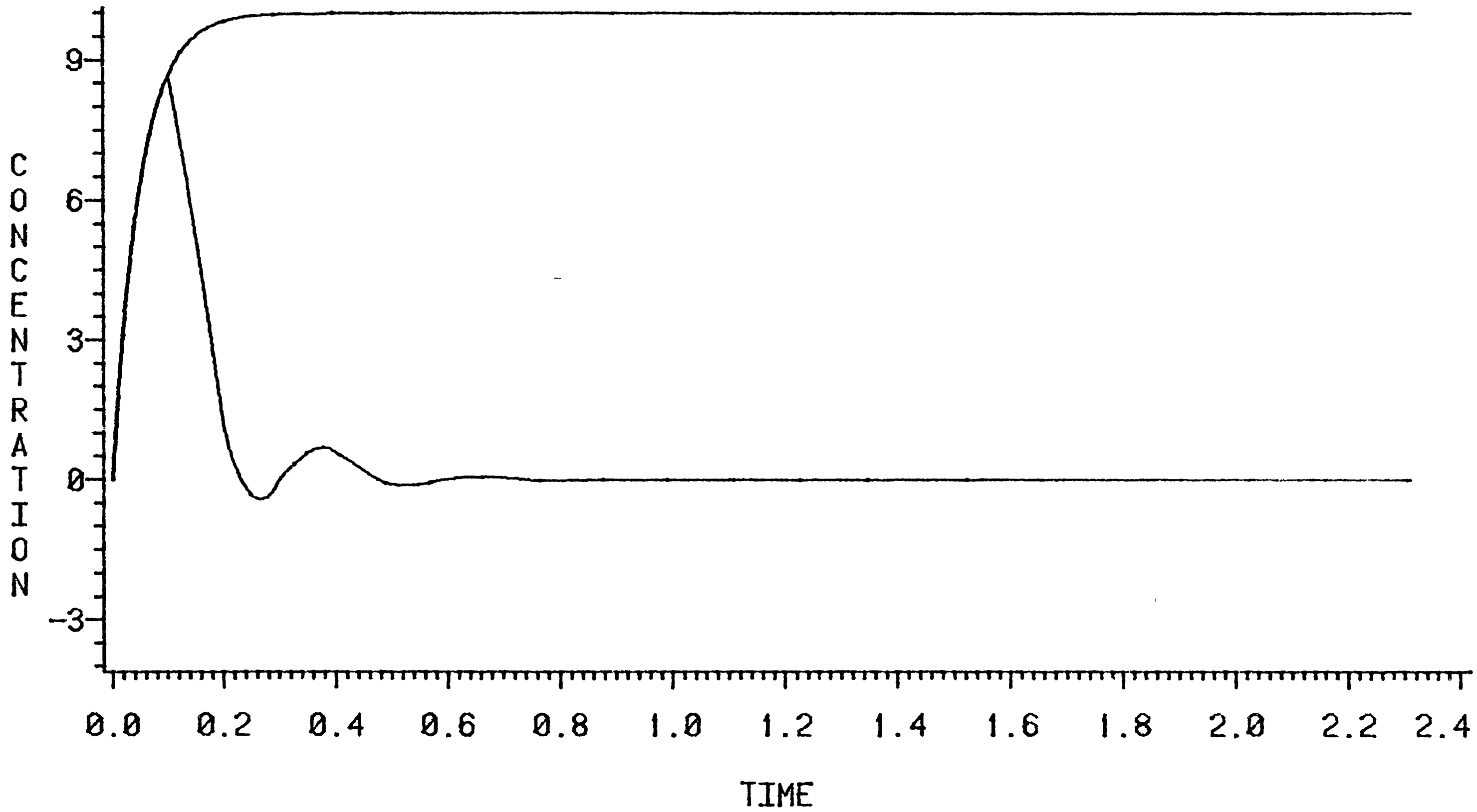


Figure 31. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.05$

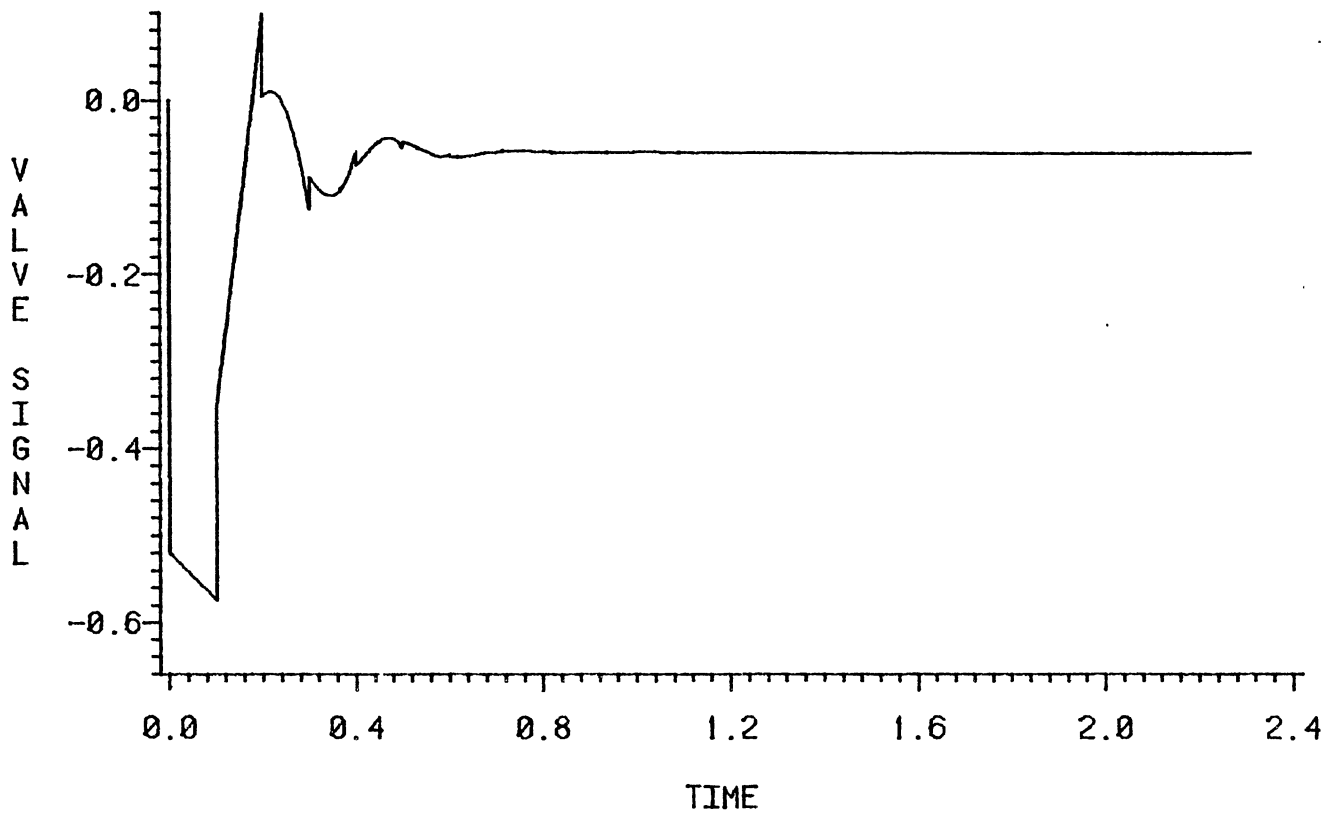


Figure 32. Valve Signal, PID Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 0.05

0.3, 0.5, 1.0, 2.0, 4.0, and 7.0.

Considering that two figures are required to present the results of a single sample application the total number of figures becomes 2 x 2 x 3 x 8 = 96 figures.

The setpoint disturbance was included with the PI controller applications because a setpoint disturbance would require the same tuning constants as a disturbance in load variable xb with $\tau_2/\tau_1 = 0$.

CHAPTER VI

DISCUSSION

Optimum Tuning Constants

The optimum PI controller tuning constants found during this study are illustrated graphically in Figures 13 and 14. The following observations may be made concerning these results. The effect of disturbance dynamics on the proportional gain K_C is most pronounced for θ_d/τ_1 ratios less than or equal to 2.0. For these curves the maximum K_C value occurs between τ_2/τ_1 values of 0.3 and 0.5. If the control system designer were to attempt to set controller settings based on the previous correlations of Rovira et al. [12] and Lopez et al. [8] he might attempt some kind of interpolation procedure. Rovira tuning constants correspond to the case where $\tau_2/\tau_1 = 0$ and Lopez tuning constants correspond to the case where $\tau_2/\tau_1 = 1.0$. The results illustrated in Figure 13 show that this interpolation approach would give less than optimum results due to the location of the maximum K_C values.

The effect of disturbance dynamics on the optimum PI controller integral time tuning constant τ_I is illustrated in Figure 14. These results show that integral action expressed a the ratio τ_1/τ_I should be increased as τ_2/τ_1 increases through most of the range of τ_2/τ_1 values examined. However, as τ_2/τ_1 approaches the value of seven the τ_1/τ_I ratio appears to be asymptotically approaching a maximum value.

The normalized IAE values for PI controller runs with optimum tuning constants are presented in Figures 15 and 16. These results show that the IAE increases with increases in θ_d/τ_1, but decreases with increases in τ_2/τ_1.

Similar trends are observed in the PID controller results presented in Figures 17 to 21. In the case of the derivative time tuning constant shown in Figure 19, the optimum value of τ_D increases almost linearly with τ_2/τ_1.

In order to evaluate the relative improvement attainable using the optimum tuning constants developed during this work a series of runs were performed comparing IAE values obtained by the present method and by the previous methods of Rovira et al. [12] and Lopez et al. [8]. The results of these calculations are presented as IAE ratio graphs given in Appendix D. These results show that the previous methods of Rovira and Lopez work well for the particular type of disturbance for which they were designed ($\tau_2/\tau_1 = 0$ for Rovira, $\tau_2/\tau_1 = 1.0$ for Lopez). However, significant improvement can be obtained by using the methods developed in this research for disturbances with dynamics different from those for which the previous methods were designed. The greatest improvement occurs for τ_2/τ_1 values larger than two. In this range the IAE values obtained by the present method are typically 25 to 85 percent of the IAE value obtained by previous methods.

Curve Fitting Results

A comparison of the optimum tuning constants graphs given in Chapter III with the graphs based on the fitted equations given in Chapter IV shows that the fitted equations approximate the optimization data with good accuracy. However, there are a few cases where the control systems designer may prefer to use the graphs presented in Chapter III rather than the fitted equations. For example, the fitted equation for the PID controller $K_C \cdot K_L$ value with θ_d/τ_1 = 0.075 gives a low standard deviation (0.272) when all τ_2/τ_1 values are considered. However, for τ_2/τ_1 values less than two a comparison of the graphs shows significant differences between this fitted equation and the optimization data. With this caution in mind, that visual comparison of equations and data should be made over the local region of interest, the control systems designer should be able to make beneficial use of the curve fitting results.

Sample Applications Results

The sample applications results illustrated in Appendix C show the effects of process dead time and disturbance dynamics on optimum control system performance. These graphs may be used to compare the response of the controlled variable x3 when using either a PI controller or a PID controller. Such a comparison shows that the controlled variable returns to setpoint quicker when under PID control. The magnitude of the load variable xb was 100 mg/L for the PI controller runs and 10 mg/L for the PID controller runs. The final change in valve signal is another way that load magnitude may be determined. In the case of the PI controller a load change in xb = 100 mg/L corresponds to a final change in valve signal va = - 0.6 psi (after the controlled variable x3 has returned to zero). In the case of the PID controller a load change xb = 10 mg/L corresponds to a final change in valve signal va = -0.06 psi. In general the steady state change in valve signal va may be related to the load magnitude change in xb by using the valve gain kv:

$$\Delta\ va = -\frac{\Delta xb}{kv} \qquad [15]$$

In this study the valve was considered to operate using a pneumatic signal in the range of 3 to 15 psig. The maximum acid flow available was 2000 mg/min when the valve was fully open. At normal operating conditions the valve would be half open supplying 1000 mg/min of acid to the mixing tank. In terms of deviation variables the valve signal could increase +6 psi before reaching its upper constraint, and decrease -6 psi before reaching its lower constraint.

The optimum tuning constants determined by this study are applicable to load changes that are small enough such that the control valve does not reach a constraint during a response to a disturbance. In order to determine the maximum load change allowable it is useful to define a load fraction:

$$\text{Load Fraction} = \frac{\text{Steady State Change in Valve Signal}}{\text{Available Change in Valve Signal}} \qquad [16]$$

In the case of a positive change in xb the control system will respond with a negative change in valve signal. The maximum available change in the valve signal is -6 psi. In the case of the PI controller the load magnitude may be expressed as a load fraction = - 0.6/-6.0 = 0.1. In the case of the PID controller the load magnitude may be expressed as a load fraction = - 0.06/- 6.0 = 0.01.

A review of the figures in Appendix C shows that the load fractions used in this study were small enough such that throughout all control system responses the valve signal va never reached -6.0. The maximum allowable load fraction (with va always remaining in the range $-6.0 \leq va \leq 6.0$) may be determined by the following equation:

$$\text{Maximum Load Fraction} = \text{Sample Application Load Fraction} \times \frac{-6.0}{(\text{Minimum Valve Signal Observed in Sample Application})} \qquad [17]$$

The maximum allowable load fraction (preventing valve saturation) has been calculated for each of the runs presented in Appendix C. The results are given in Figure 33 for the PI controller and in Figure 34 for the PID controller. These figures show that under similar conditions a PID controller is more sensitive to load fraction than is a PI controller. For either controller the allowable load fraction decreases with decreases in both θ_d/τ_1 and τ_2/τ_1.

While previous workers have considered isolated cases of the effect of disturbance dynamics ($\tau_2/\tau_1 = 0, 1$), the author is not aware of any previous research on the effect of disturbance magnitude. The concept of allowable load fraction addresses this practical problem in an effective manner.

A final comment can be made concerning the sample applications given in Appendix C. There is a marked difference in the general shape of the valve signal response for the cases of the PI controller and the PID controller. In the case of the PI controller the valve response is relatively gradual and continuous as it moves to correct a load disturbance. In the case of the PID controller there are a number of sharp discontinuities in the valve signal response. In order to obtain the improvement predicted for a PID controller it may be necessary to install a high performance valve capable of quick and accurate response to the valve signal.

It should be noted that the discontinuities in the PID controller valve response occur at intervals equal to the process dead time θ_d. The discontinuities are not due to the numerical technique employed in this study. This has been checked by repeating the sample application runs with different values of the Euler integration step size. When the step size is either doubled or halved the response of the valve signal is essentially identical.

A more serious finding occurs in sample applications using the PID controller with $\tau_2/\tau_1 = 7.0$ and $\theta_d/\tau_1 = 0.5$ or 1.0 (see Figures 113, 114, 130, and 131). In these runs the valve response appears to be diverging, indicating that the control system is gradually going unstable. For these cases the Bode criterion was used to check for system stability. This criterian would indicate that the optimized tuning constants bring the system very close to the stability limit (amplitude ratio = 1.0 at - 180 degrees phase lag) in these particular cases. Rechecking the system stability for other runs led to the following conclusion. The PID controller optimum tuning constants are not recommended for: $\theta_d/\tau_1 = 0.4$ and $\tau_2/\tau_1 > 6.5$, $\theta_d/\tau_1 = 0.5$ and $\tau_2/\tau_1 > 5.5$, $\theta_d/\tau_1 = 0.75$ and $\tau_2/\tau_1 > 5.0$, $\theta_d/\tau_1 = 1.0$ and $\tau_2/\tau_1 > 4.5$.

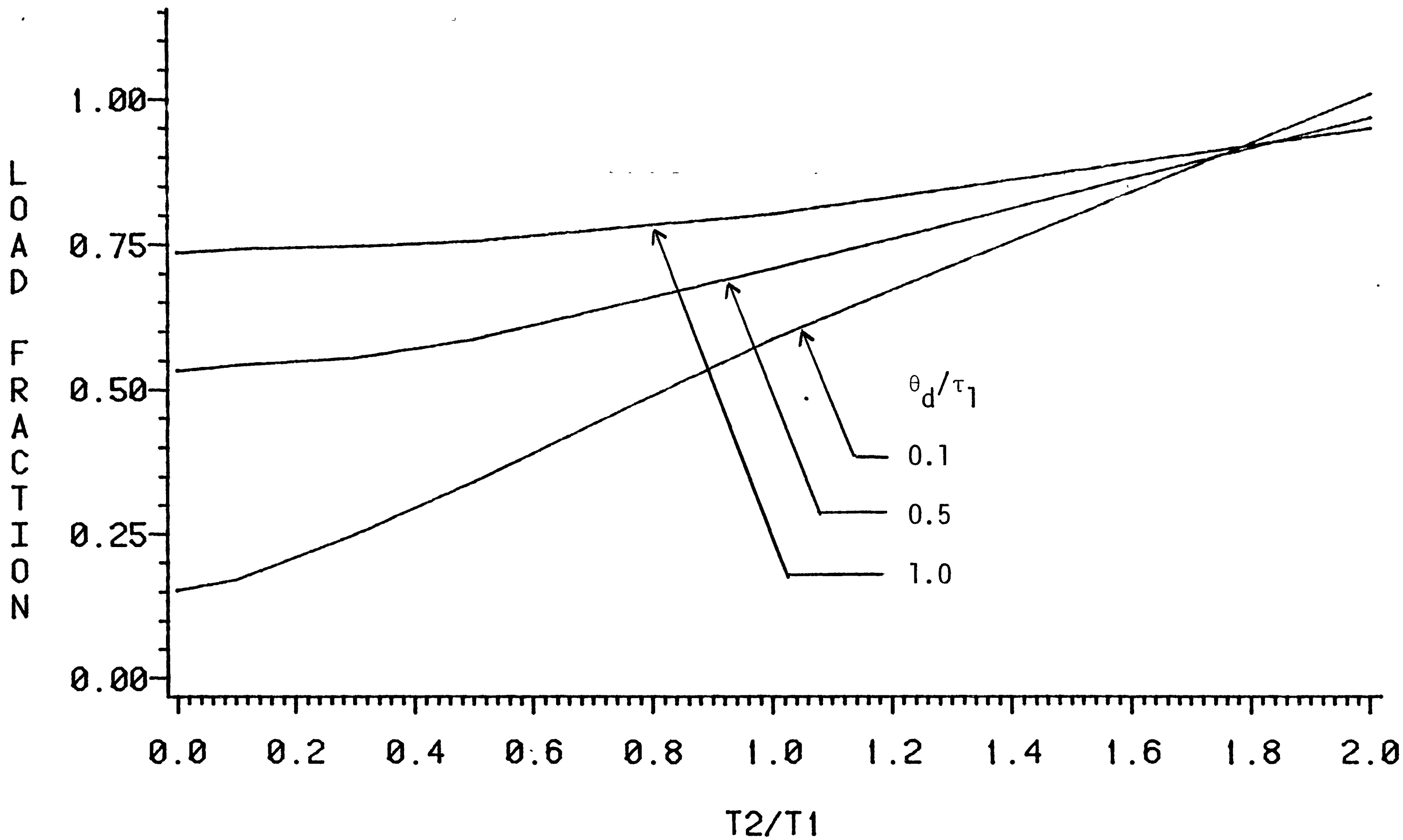


Figure 33. Maximum Allowable Load Fraction (Preventing Valve Saturation) for a PI Controller Using Optimum Tuning Constants

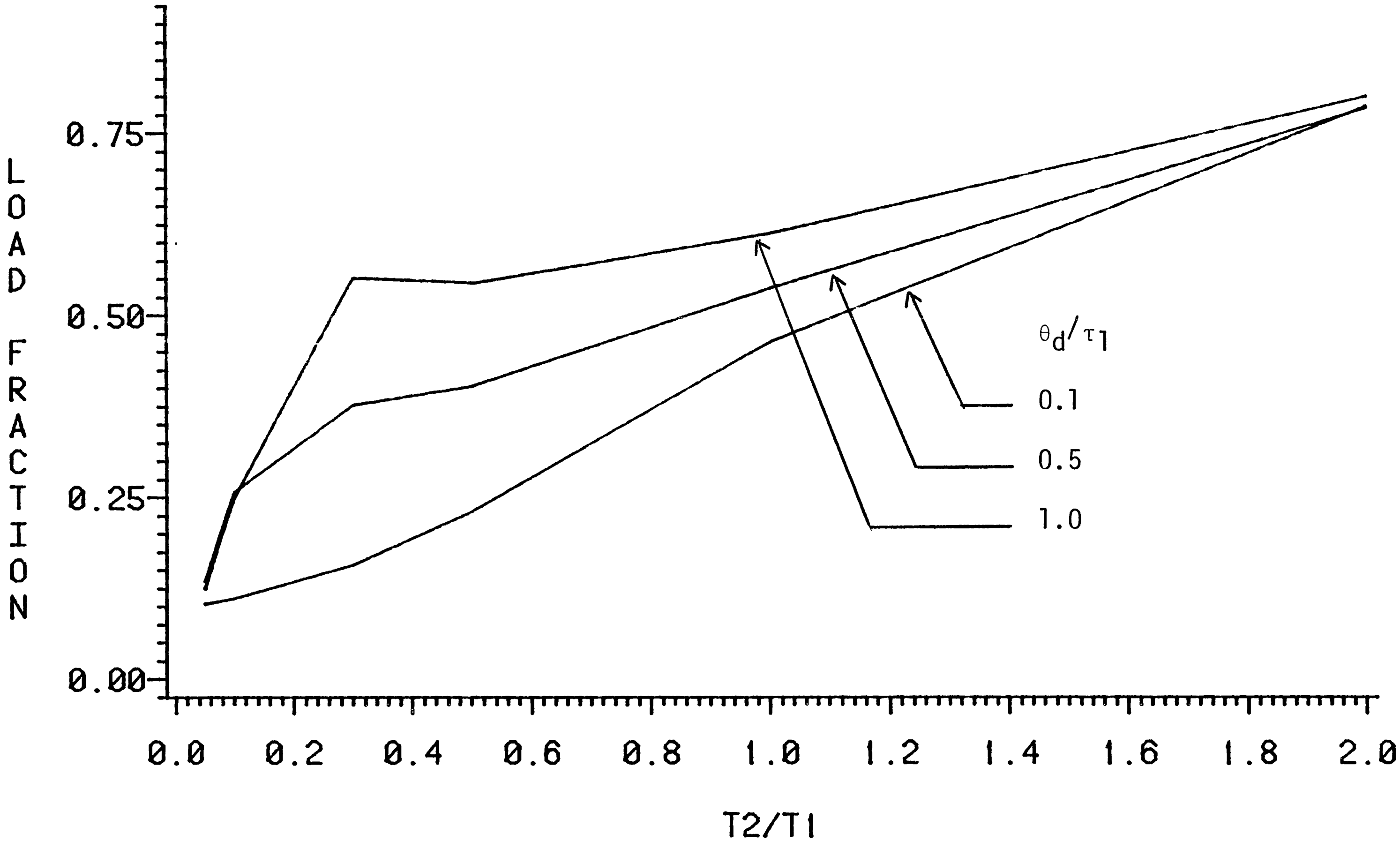


Figure 34. Maximum Allowable Load Fraction (Preventing Valve Saturation) for a PID Controller Using Optimum Tuning Constants

CHAPTER VII

CONCLUSIONS

The following conclusions may be drawn as a result of this study:

1. A computer simulation approach has been used to develop a method of finding optimum tuning constants for proportional-integral and proportional-integral-derivative controllers. Consideration of the effect of disturbance dynamics is a unique feature of this method. The method is applicable to systems that allow the process to be modeled as first order plus dead time; with gain K_p, time constant τ_1, and dead time θ_d. The disturbance is modeled as first order, with time constant τ_2, and enters the loop just before the sensor.

2. Optimum tuning constants are reported graphically in Chapter III and in the form of correlations in Chapter IV. The normalized tuning constants for PI and PID controllers are reported as a function of the ratios θ_d/τ_1 and τ_2/τ_1. The method is applicable for $0.05 \leq \theta_d/\tau_1 \leq 1.0$ and $0 \leq \tau_2/\tau_1 \leq 7.0$.

3. In addition to the tuning constants, the normalized integral of the absolute value of the error (used as the optimization criterion) associated with the optimum tuning constants is reported graphically and in the form of correlations.

4. The validity of the approach has been reinforced by demonstrating that the results of previous workers (Rovira et al. - $\tau_2/\tau_1 = 0$ and Lopez et al. - $\tau_2/\tau_1 = 1.0$) may be considered as subsets of the results

found by the present method. This point is described in Chapter VI and Appendix D.

5. The integral of the absolute value of the error (IAE) obtainable by the present method is compared in Appendix D to the IAE obtainable by the methods of Rovira and Lopez. The improvement found with the present method is greatest for $\tau_2/\tau_1 > 2$. In this range the present method IAE is typically 25 to 85 percent of the IAE found with previous workers methods.

6. Sample applications were performed using the optimum tuning constants found by the present method. The results were presented in Appendix C. Graphs were provided showing the time response of the disturbance variable, the controlled variable, and the valve signal. The results show that the PID controller is able to obtain better controlled variable response than the PI controller. However, the response of the PID controller valve signal is sharply discontinuous (discontinuities appearing at intervals of θ_d). On the other hand, the PI controller valve signal shows a smooth continuous response.

7. The effect of disturbance magnitude in the presence of manipulated variable (valve) constraints was considered. For load magnitudes below a certain limiting value the response of the valve signal always remains within the allowable range and does not become saturated. Optimum tuning constants reported in this study are applicable to load magnitudes below this limiting value. The limiting value of the load magnitude that will give a saturated valve was calculated for the sample applications runs presented in Appendix C. These results were reported in the form of maximum load fractions given in Chapter VI. The PID controller was shown to be more sensitive to load fraction than was the PI controller.

8. Certain sample applications runs (see Figures 113, 114, 130, and 131) indicate that it may be possible for systems near the Bode criterion stability limit to have "optimum" tuning constants and yet appear to be moving towards divergent response. Accordingly the following PID controller tuning constants found by this study are not recommended when: $\theta_d/\tau_1 = 0.4$ and $\tau_2/\tau_1 > 6.5$, $\theta_d/\tau_1 = 0.5$ and $\tau_2/\tau_1 > 5.5$, $\theta_d/\tau_1 = 0.75$ and $\tau_2/\tau_1 > 5.0$, $\theta_d/\tau_1 = 1.0$ and $\tau_2/\tau_1 > 4.5$.

SELECTED BIBLIOGRAPHY

1. Anderson, N. A. "Step-Analysis Method of Finding Time Constant." Instruments and Control Systems, Vol. 36 (1963), p. 130.

2. Caceci, M. S., and W. P. Cacheris. "Fitting Curves to Data." Bytes, Vol. 9 (1984), pp. 340-362.

3. Clark, R. and S. Koehler. The UCSD Pascal Handbook. New Jersey: Prentice-Hall, Inc., 1982.

4. Cohen, G. H., and G. A. Coon. "Theoretical Consideration of Retarded Control." Instruments and Control Systems, Vol. 36 (1963), p. 130.

5. Deshpande, P. B., and R. H. Ash. Elements of Computer Process Control With Advanced Control Applications. North Carolina: Instrument Society of America, 1981.

6. Kuester, J. L. and J. H. Mize. Optimization Techniques. New York: McGraw-Hill Co., 1973.

7. Lopez, A. M. "Optimization of System Response (Volumes I and II)." (Unpub. Ph.D. dissertation, Louisiana State University, 1967.)

8. Lopez, A. M., J. A. Miller, C. L. Smith, and P. W. Smith. "Tuning Controllers with Error-Integral Criteria." Instrumentation Technology, Vol. 14 (1967), pp. 57-62.

9. Luyben, W. L. Process Modeling, Simulation, and Control for Chemical Engineering. New York: McGraw-Hill Co., 1973.

10. Nedler, J. A., and R. Mead. "A Simple Method for Function Minimization." Computer J., Vol. 7 (1965), p. 308.

11. Rosenbrock, H. H. "An Automatic Method for Finding the Greatest or Least Value of a Function." Computing J., Vol. 3 (1960), pp. 175-184.

12. Rovira, A. A. "Control Algorithms, Tuning, and Adaptive Gain Tuning in Process Control." (Unpub. Ph.D. dissertation, Louisiana State University, 1981.)

13. Rovira, A. A., P. W. Murrill, and C. L. Smith. "Tuning Controllers for Setpoint Changes." Instrumentation Technology, Vol. 16 (1969), pp. 67-69.

14. SAS/GRAPH User's Guide. North Carolina, SAS Institute Inc., 1982.

15. Sood, M. "Tuning Proportional-Integral Controllers for Random Load Changes." (Unpub. M.S. Thesis, University of Mississippi, 1972.)

16. Sood, M., and H. T. Huddleson. "Optimum Control Settings for Random Disturbances." Instrumentation Technology, Vol. 20 (1973), pp. 47-49.

17. Sood, M. and H. T. Huddleson. "Tuning PID Controllers for Random Disturbances." Instrumentation Technology, Vol. 24 (1977), pp. 61-63.

18. Sten, J. W. "Evaluating Second-Order Parameters." Instrumentation Technology, Vol. 17 (1970), p. 39.

19. Ziegler, J. G., and N. B. Nichols. "Optimal Settings for Automatic Controllers." Trans. Amer. Soc. Mech. Engrs., Vol. 64 (1942), pp. 759-768.

20. Zoss, L. M. "How to Reckon Basic Process Dynamics." Control Engineering, Vol. 3 (1956), p. 50.

APPENDIX A

OPTIMUM TUNING CONSTANTS PROGRAM

```
//U11217A  JOB (?????,XXX-XX-XXXX),'SIRIPOL',TIME=(0.34),CLASS=F,
//   MSGCLASS=X,NOTIFY=U11217A
/*PASSWORD ?
/*JOBPARM ROOM=X
//   EXEC PASCAL,REGION=512K
//PASC.SYSIN  DD  *
PROGRAM ROSEN (INPUT,OUTPUT);
 LABEL 10,20,30,40;
 CONST
  KM=3; MAXK=5000; MKAT=90; MCYC=50; NSTEP=1;
  EPSY=0.0000000001; ALPHA=3.0; BETA=0.5;
  KMM=0.004; KV=166.66667;
 VAR
  PARM:ARRAY(.1..KM.) OF REAL;
   V:ARRAY(.1..KM,1..KM.) OF REAL;
    BL:ARRAY(.1..KM,1..KM.) OF REAL;
     D:ARRAY(.1..KM.) OF REAL;
      BLEN:ARRAY(.1..KM.) OF REAL;
       EPS:ARRAY(.1..KM.) OF REAL;
        AJ:ARRAY(.1..KM.) OF REAL;
         E:ARRAY(.1..KM.) OF REAL;
        AL:ARRAY(.1..KM,1..KM.) OF REAL;
         I,II,III,J,K,L,KAT,KK1,KL:INTEGER;
         SUMO,SUMN,FBEST,SUMDIF,SUMAV,SUMAVV:REAL;
        DT : ARRAY(.1..3000.) OF REAL;
       DELTA,TAU1,TAU2,TT,TIME,THETAD:REAL;
      X1,X2,X3,X4,C,VV,VA,XA,XB,R:REAL;
     X4DOT, X2DOT:REAL;
    ER,EPAST,ERINT,ABSIE:REAL;
   G,Q,S:INTEGER;
  KC,TAUI,TAUD:REAL;
PROCEDURE DATA;
 BEGIN
  WRITELN (OUTPUT,' CASE OF PID WITH LOPEZ STARTING VALUE');
  WRITELN (OUTPUT,' TT=18.582 THETAD=1.0 TAU1=1.0 TAU2=7.0');
  READLN  (INPUT, KC, TAUI, TAUD);
  WRITELN (OUTPUT,' STARTING VALUE:PROPORTIONAL TUNING = ',KC:10:5);
  WRITELN (OUTPUT,' STARTING VALUE: INTEGRAL TUNING = ',TAUI:10:5);
  WRITELN (OUTPUT,' STARTING VALUE: DERIVATIVE TUNING = ',TAUD:10:5);
  READLN  (INPUT, R, XA, XB);
  WRITELN (OUTPUT,' SET POINT = ',R:10:5);
  WRITELN (OUTPUT,' XA = ',XA:10:5);
  WRITELN (OUTPUT,' XB = ',XB:10:7);
  READLN  (INPUT, DELTA, TT, THETAD);
  WRITELN (OUTPUT,' DELTA = ',DELTA:10:5);
  WRITELN (OUTPUT,' TOTAL TIME = ',TT:10:5);
  WRITELN (OUTPUT,' THETAD = ',THETAD:10:5);
  READLN  (INPUT, TAU1, TAU2);
  WRITELN (OUTPUT,' TAU1 = ',TAU1:10:5);
  WRITELN (OUTPUT,' TAU2 = ',TAU2:10:5);
 END;
PROCEDURE PRINT (DUM1,DUM2,DUM3,DUM4,DUM5,DUM6:REAL);
 BEGIN
  WRITELN (OUTPUT,' NO OF STAGES = ',DUM1);
  WRITELN (OUTPUT,' NO OF FUNCTION EVALUATION = ',DUM2);
  WRITELN (OUTPUT,' FINAL VALUE OF FUNCTION = ',DUM3:16:8);
  WRITELN (OUTPUT,' X(1) = ',DUM4:16:8);
  WRITELN (OUTPUT,' X(2) = ',DUM5:16:8);
  WRITELN (OUTPUT,' X(3) = ',DUM6:16:8);
  HALT;
 END;
FUNCTION OBJECT (KCC,TAUII,TAUDD:REAL):REAL;
 VAR I : INTEGER;
  BEGIN
   EPAST := 0.0; ERINT := 0.0; ABSIE := 0.0;
    TIME := 0.0;
     FOR I := 1 TO S DO DT(.I.) := 0.0;
      G := S; Q := 1; C := 0.0; X2 := 0.0; X3 := 0.0; X4 := 0.0;
       WHILE TIME < TT DO
        BEGIN
         C := KMM*X3;
          ER := R - C;
           VA := KCC*(ER + ERINT/TAUII + (ER - EPAST)*TAUDD/DELTA);
            VV := VA;
```

```
                  IF VA <= -6.0 THEN VA := -6.0;
                  IF VA >= 6.0 THEN VA := 6.0;
                  X1 := XA + VA*KV;
                  X4DOT := (XB - X4)/TAU2;
                  X2DOT := (X1 - X2)/TAU1;
                  TIME := TIME + DELTA;
                  IF (VV < 6.1) AND (VV > -6.1) THEN
                  ERINT := ERINT + ER*DELTA;
                  ABSIE := ABSIE + ABS(ER*DELTA);
                  EPAST := ER;
                  X2 := X2 + X2DOT*DELTA;
                  DT(.G.):=X2;
                  X4 := X4 + X4DOT*DELTA;
                 X3 := DT(.Q.) + X4;
               G := G + 1; Q := Q + 1;
              IF G > S THEN G := 1;
             IF Q > S THEN Q := 1;
            END;
         OBJECT := ABSIE;
       WRITELN (OUTPUT,' ABSIE = ',ABSIE,' VA=',VA:10:5);
     END;
     BEGIN
            DATA;
            PARM(.1.):=KC;
            PARM(.2.):=TAUI;
            PARM(.3.):=TAUD;
            EPS(.1.):=0.1;
            EPS(.2.):=0.02;
            EPS(.3.):=0.005;
            S := ROUND(THETAD / DELTA) + 1;
            KAT:=1;
            FOR I:=1 TO KM DO
             BEGIN
               FOR J:=1 TO KM DO
                BEGIN
                 V(.I,J.):=0.0;
                  IF I=J THEN
                   V(.I,J.):=1.0;
                END;
             END;
            SUMN:=OBJECT(PARM(.1.),PARM(.2.),PARM(.3.));
            SUMO:=SUMN;
            KK1:=1;
            IF NSTEP <> 1 THEN
             BEGIN
               FOR I:=1 TO KM DO
               E(.I.):=EPS(.I.);
             END;
     10:    FOR I:=1 TO KM DO
             BEGIN
              FBEST:=SUMN;
               AJ(.I.):=2.0;
                IF NSTEP=1 THEN
               E(.I.):=EPS(.I.);
              D(.I.):=0.0;
             END;
            III:=0;
     20:    III:=III + 1;
     30:    I:=1;
     40:    FOR J:=1 TO KM DO
             PARM(.J.):=PARM(.J.) + E(.I.)*V(.I,J.);
            SUMN:=OBJECT(PARM(.1.),PARM(.2.),PARM(.3.));
            KAT:=KAT + 1;
            SUMDIF:=FBEST - SUMN;
            IF (ABS(SUMDIF) - EPSY) <= 0.0 THEN
            PRINT (KK1,KAT,SUMO,PARM(.1.),PARM(.2.),PARM(.3.));
            IF (KAT - MAXK) >= 0 THEN
              BEGIN
               WRITELN (OUTPUT,' MAXK EXCEEDED');
               PRINT (KK1,KAT,SUMO,PARM(.1.),PARM(.2.),PARM(.3.));
              END;
            IF (SUMN - SUMO) > 0.0 THEN
              BEGIN
               FOR J:=1 TO KM DO
                PARM(.J.):=PARM(.J.) - E(.I.)*V(.I,J.);
```

```
   E(.I.):=-BETA*E(.I.);
  IF (AJ(.I.) - 1.5) < 0.0 THEN AJ(.I.):=0.0;
 END
ELSE
 BEGIN
  D(.I.):=D(.I.) + E(.I.);
   E(.I.):=ALPHA*E(.I.);
   SUMO:=SUMN;
  IF (AJ(.I.) -1.5 ) > 0.0 THEN AJ(.I.):=1.0;
 END;
 FOR J:=1 TO KM DO
  IF (AJ(.J.) - 0.5) > 0.0 THEN
   BEGIN
    IF (I - KM) <> 0 THEN
     BEGIN
      I:=I + 1;
      GOTO 40;
     END
   ELSE
    BEGIN
     FOR K:=1 TO KM DO
      IF (AJ(.K.) - 2.0) < 0.0 THEN GOTO 30;
     IF (III - MCYC) < 0 THEN GOTO 20
   ELSE
     BEGIN
      WRITELN (OUTPUT,' MCYC EXCEEDED');
      PRINT (KK1,KAT,SUMO,PARM(.1.),PARM(.2.),PARM(.3.));
     END;
    END;
   END;
 FOR I:=1 TO KM DO
 FOR J:=1 TO KM DO
 AL(.I,J.):=0.0;
 WRITELN (OUTPUT,' STAGE NO. = ',KK1:3);
 WRITELN (OUTPUT,' FUNCTION = ',SUMO:16:8);
 FOR I:=1 TO KM DO
 WRITELN (OUTPUT,' X(',I,') = ',PARM(.I.):16:8);
 FOR I:=1 TO KM DO
  BEGIN
   KL:=I;
   FOR J:=1 TO KM DO
    BEGIN
     FOR K:=KL TO KM DO
      AL(.I,J.):=D(.K.)*V(.K,J.) + AL(.I,J.);
     BL(.I,J.):=AL(.I,J.);
    END;
  END;
  BLEN(.1.):=0.0;
  FOR K:=1 TO KM DO
   BLEN(.1.):=BLEN(.1.) + BL(.1,K.)*BL(.1,K.);
    BLEN(.1.):=SQRT(BLEN(.1.));
   FOR J:=1 TO KM DO
 V(.1,J.):=BL(.1,J.) / BLEN(.1.);
  FOR I:=2 TO KM DO
   BEGIN
    II:=I - 1;
     FOR J:= 1 TO KM DO
      BEGIN
       SUMAVV:=0.0;
        FOR K:=1 TO II DO
         BEGIN
          SUMAV:=0.0;
           FOR L:=1 TO KM DO
            SUMAV:=SUMAV + AL(.I,L.)*V(.K,L.);
           SUMAVV:=SUMAV*V(.K,J.) + SUMAVV;
         END;
        BL(.I,J.):=AL(.I,J.) - SUMAVV;
      END;
   END;
  FOR I:=2 TO KM DO
   BEGIN
    BLEN(.I.):=0.0;
     FOR J:=1 TO KM DO
      BLEN(.I.):=BLEN(.I.) + BL(.I,J.)*BL(.I,J.);
      BLEN(.I.):=SQRT(BLEN(.I.));
```

```
              FOR K:=1 TO KM DO
             V(.I,K.):=BL(.I,K.) / BLEN(.I.);
            END;
           KK1:=KK1 + 1;
           IF (KK1 - MKAT) < 0 THEN GOTO 10;
   END.
/*
//GO.SYSIN  DD  *
2.15249 1.13895 0.48200
0.0 0.0 10.0
0.005 18.582 1.0
1.0 7.0
/*
```

APPENDIX B

CURVE FITTING PROGRAM

```
//U11217A  JOB (?????,XXX-XX-XXXX),'SIRIPOL',TIME=(0,5),CLASS=A,
//   MSGCLASS=X,NOTIFY=U11217A
/*PASSWORD ?
/*JOBPARM ROOM=X
//   EXEC PASCAL,REGION=512K
//PASC.SYSIN  DD  *
PROGRAM SIMP (INPUT,OUTPUT);
 CONST M      =   5;        (* NUMBER OF PARAMETERS TO FIT *)
       NVPP   =   6;        (* TOTAL NUMBER OF VARS. PER DATA POINT *)
       N      =   6;        (* M+1 DIMENSION *)
       MNP    = 200;        (* MAX. NUMBER OF DATA POINT *)
       ALPHA  = 1.0;        (* REFLECTION COEFFICIENT *)
       BETA   = 0.5;        (* CONTRACTION COEFFICIENT *)
       GAMMA  = 2.0;        (* EXPANSION COEFFICIENT *)
       LW     =   4;        (* WIDTH OF LINE IN DATA FIELDS + 1 *)
       ROOT2 =   1.414214;
 TYPE  VECTOR     =   ARRAY(.1..N.) OF REAL;
       DATAROW    =   ARRAY(.1..NVPP.) OF REAL;
       INDEX      =   0..255;
 VAR   DONE       : BOOLEAN;           (*CONVERGENCE*)
       I,J        : INDEX;
       H,L        : ARRAY(.1..N.) OF INDEX;
       NP,
       MAXITER,                        (*MAX. NUMBER OF ITERATION*)
       NITER      : INTEGER;           (*NUMBER OF ITERATION*)
       NEXT,                           (*NEW VERTEX TO BE TESTED*)
       CENTER,                         (*CENTER OF HYPERPLANE*)
       MEAN, ERROR,
       MAXERR,                         (*MAX. ERROR ACCEPTED*)
       P,Q,                            (*TO COMPUTE FIRST SIMPLEX*)
       STEP       : VECTOR;            (*INPUT STARING STEP*)
       SIMP       : ARRAY(.1..N.) OF VECTOR;       (*THE SIMPLEX*)
       DATA       : ARRAY(.1..MNP.) OF DATAROW;    (*THE DATA*)
FUNCTION F (X : VECTOR; D : DATAROW) : REAL;
 BEGIN
   F:=X(.1.)-EXP(-X(.2.)*D(.1.))*
      (X(.3.)*COS(X(.4.)*D(.1.)) + X(.5.)*SIN(X(.4.)*D(.1.)));
 END;
PROCEDURE SUM_OF_RESIDUALS (VAR X : VECTOR);
 VAR I : INDEX;
  BEGIN
   X(.N.) := 0.0;
    FOR I := 1 TO 18 DO
     BEGIN
      X(.N.):=X(.N.)+SQR(F(X,DATA(.I.)) - DATA(.I,3.))
     END
  END;
PROCEDURE ENTER;
  VAR I,J,NP:INTEGER;
   BEGIN
   WRITELN (OUTPUT,' TO FIND KCKL CASE OF PID,TT=18.582,THETAD=1.0');
   WRITELN (OUTPUT,' MODEL IS KCKL=A-EXP(-BX)*(C*COS(DX)+E*SIN(DX))');
   READ (INPUT,MAXITER);
   WRITELN (OUTPUT,' MAX NUMBER OF ITERATIONS IS := ',MAXITER:5);
   WRITELN (OUTPUT,' START COORD.:');
   FOR I := 1 TO M DO
    BEGIN
     READ (INPUT,SIMP(.1,I.));
        IF (I MOD LW) = 0 THEN WRITELN (OUTPUT);
     WRITE (OUTPUT,SIMP(.1,I.));
    END;
   WRITELN (OUTPUT);
   WRITE (OUTPUT,' START STEPS: ');
   FOR I := 1 TO M DO
    BEGIN
     READ (INPUT,STEP(.I.));
        IF (I MOD LW) = 0 THEN WRITELN (OUTPUT);
     WRITE (OUTPUT,STEP(.I.))
    END;
   WRITELN (OUTPUT);
   WRITE (OUTPUT,' MAX. ERRORS: ');
    FOR I := 1 TO N DO
    BEGIN
     READ (INPUT,MAXERR(.I.));
```

```
          IF (I MOD LW) = 0 THEN WRITELN (OUTPUT);
       WRITE (OUTPUT,MAXERR(.I.))
      END;
     WRITELN (OUTPUT);
     WRITELN (OUTPUT,' DATA:');
     WRITELN (OUTPUT,' X':14,'KCKL':14);
     FOR NP:=1 TO 18 DO
      BEGIN
       WRITE (OUTPUT,' #',NP:3);
        FOR J := 1 TO NVPP DO
         BEGIN
            READ (INPUT,DATA(.NP,J.));
            WRITE (OUTPUT,DATA(.NP,J ));
         END;
        WRITELN (OUTPUT);
      END
     END;
  PROCEDURE REPORT;
   VAR KCKL,DKCKL,
        SIGMA      :      REAL;
        I,J        :       INTEGER;
    BEGIN
     WRITELN (OUTPUT,' PROGRAM EXITED AFTER',NITER:5,'ITERATIONS');
     WRITELN (OUTPUT,' THE FINAL SIMPLEX IS');
     FOR J := 1 TO N DO
      BEGIN
       FOR I := 1 TO N DO
        BEGIN
           IF (I MOD LW) = 0 THEN WRITELN (OUTPUT);
           WRITE (OUTPUT,SIMP(.J,I.):10)
        END;
       WRITELN (OUTPUT);
      END;
     WRITELN (OUTPUT,' THE MEAN IS');
     FOR I := 1 TO N DO
      BEGIN
         IF (I MOD LW) = 0 THEN WRITELN (OUTPUT);
         WRITE (OUTPUT,MEAN(.I.))
      END;
     WRITELN (OUTPUT);
     WRITELN (OUTPUT,' THE ESTIMATED FRACTIONAL ERROR IS ');
     FOR I := 1 TO N DO
      BEGIN
         IF (I MOD LW) = 0 THEN WRITELN (OUTPUT);
         WRITE (OUTPUT,ERROR(.I.));
      END;
     WRITELN (OUTPUT);
     WRITELN (OUTPUT,' #':4,'X':10,'KCKL':15,'KCKL"':15,'DKCKL':15);
     SIGMA := 0.0;
     FOR I := 1 TO 18 DO
      BEGIN
       KCKL := F(MEAN,DATA(.I.));
       DKCKL := DATA(.I,3.) - KCKL;
       SIGMA := SIGMA + SQR(DKCKL);
       WRITELN (I:4,DATA(.I,1.):15,DATA(.I,3.):15,KCKL:15,DKCKL:15);
      END;
     SIGMA := SQRT(SIGMA / 18);
     WRITELN (OUTPUT,' THE STANDARD DEVIATION IS',SIGMA);
     SIGMA := SIGMA / SQRT(18 - M);
     WRITE (OUTPUT,' THE ESTIMATED ERROR OF THE');
     WRITELN (OUTPUT,' FUNCTION IS',SIGMA);
    END;
  PROCEDURE FIRST;
    VAR I,J:INTEGER;
      BEGIN
       WRITELN (OUTPUT,' STARTING SIMPLEX');
        FOR J := 1 TO N DO
         BEGIN
          WRITE (OUTPUT,' SIMP(',J:1,')');
          FOR I := 1 TO N DO
           BEGIN
            IF (I MOD LW) = 0 THEN WRITELN (OUTPUT);
            WRITELN (OUTPUT,SIMP(.J,I.))
           END;
          WRITELN (OUTPUT)
```

```
        END;
      WRITELN (OUTPUT)
     END;
PROCEDURE NEW_VERTEX;
  VAR I : INTEGER;
   BEGIN
    FOR I := 1 TO N DO
     BEGIN
      SIMP(.H(.N.),I.) := NEXT(.I.);
     END;
   END;
PROCEDURE ORDER;
  VAR I,J   :   INDEX;
   BEGIN
    FOR J:= 1 TO N DO
     BEGIN
      FOR I := 1 TO N DO
       BEGIN
        IF SIMP(.I,J.) < SIMP(.L(.J.),J.) THEN L(.J.) := I;
        IF SIMP(.I,J.) > SIMP(.H(.J.),J.) THEN H(.J.) := I;
       END
     END
   END;
BEGIN
  ENTER;
  SUM_OF_RESIDUALS(SIMP(.1.));
  FOR I := 1 TO M DO
   BEGIN
    P(.I.) := STEP(.I.) * (SQRT(N) + M - 1) / (M * ROOT2);
    Q(.I.) := STEP(.I.) * (SQRT(N) - 1 ) / (M * ROOT2)
   END;
  FOR I := 2 TO N DO
   BEGIN
    FOR J := 1 TO M DO SIMP(.I,J.) := SIMP(.1,J.) + Q(.J.);
    SIMP(.I,(I-1).) := SIMP(.1,(I-1).) + P(.(I-1).);
    SUM_OF_RESIDUALS(SIMP(.I.))
   END;
  FOR I := 1 TO N DO
   BEGIN
    L(.I.) := 1; H(.I.) := 1
   END;
  ORDER;
  FIRST;
  NITER := 0;
  REPEAT
  DONE := TRUE;
  NITER := SUCC(NITER);
  FOR I := 1 TO N DO CENTER(.I.) := 0.0;
  FOR I := 1 TO N DO
   IF I <> H(.N.) THEN
    FOR J := 1 TO M DO
     CENTER(.J.) := CENTER(.J.) + SIMP(.I,J.);
  FOR I := 1 TO N DO
   BEGIN
    CENTER(.I.) := CENTER(.I.) / M;
    NEXT(.I.) := (1.0 + ALPHA)*CENTER(.I.) - ALPHA*SIMP(.H(.N.),I.)
   END;
  SUM_OF_RESIDUALS(NEXT);
  IF NEXT(.N.) <= SIMP(.L(.N.),N.) THEN
   BEGIN
    NEW_VERTEX;
     FOR I := 1 TO M DO
       NEXT(.I.) := GAMMA*SIMP(.H(.N.),I.) + (1.0 - GAMMA)*CENTER(.I.);
       SUM_OF_RESIDUALS(NEXT);
     IF NEXT(.N.) <= SIMP(.L(.N.),N.) THEN NEW_VERTEX
    END
    ELSE
    BEGIN
     IF NEXT(.N.) <= SIMP(.H(.N.),N.) THEN NEW_VERTEX
      ELSE
       BEGIN
        FOR I := 1 TO M DO
          NEXT(.I.) := BETA*SIMP(.H(.N.),I.) + (1.0 - BETA)*CENTER(.I );
          SUM_OF_RESIDUALS(NEXT);
        IF NEXT(.N.) <= SIMP(.H(.N.),N.) THEN NEW_VERTEX
```

```
         ELSE
          BEGIN
           FOR I := 1 TO N DO
            BEGIN
             FOR J := 1 TO M DO
               SIMP(.I,J.) := (SIMP(.I,J.) + SIMP(.L(.N.),J.))*BETA;
             SUM_OF_RESIDUALS(SIMP(.I.))
            END
           END
          END
       END;
       ORDER;
       FOR J := 1 TO N DO
        BEGIN
         ERROR(.J.) := (SIMP(.H(.J.),J.) - SIMP(.L(.J.),J.)) /
                       SIMP(.H(.J.),J.);
            IF DONE THEN
          IF ERROR (.J.) > MAXERR(.J.) THEN
         DONE := FALSE
        END
       UNTIL (DONE OR (NITER = MAXITER));
       FOR I := 1 TO N DO
        BEGIN
         MEAN(.I.) := 0.0;
           FOR J := 1 TO N DO
           MEAN(.I.) := MEAN(.I.) + SIMP(.J,I.);
         MEAN(.I.) := MEAN(.I.) / N
        END;
       REPORT;
END.
/*
//GO.SYSIN  DD  *
100000
0.1 0.2 0.1 0.2 0.1
0.2 0.1 0.2 0.1 0.2
1E-6 1E-6 1E-6 1E-6 1E-6 1E-6
0.05 1.0 1.09045735 0.73412517 0.36202755 1.34195495
0.10 1.0 1.09491163 0.73176159 0.36060511 1.33489132
0.30 1.0 1.17685113 0.74044594 0.37875813 1.26401612
0.50 1.0 1.28375105 0.80014963 0.39567915 1.15759400
0.70 1.0 1.36190855 0.84495412 0.40962450 1.04953215
1.00 1.0 1.44784678 0.91146500 0.41434703 0.90645150
1.50 1.0 1.53778282 1.03390894 0.42546319 0.72278022
2.00 1.0 1.59013203 1.17773911 0.45314590 0.59910007
2.50 1.0 1.61987303 1.30816817 0.49064089 0.52049807
3.00 1.0 1.63306075 1.42468300 0.52184435 0.46942470
3.50 1.0 1.63699425 1.52347841 0.54867154 0.43321135
4.00 1.0 1.63353460 1.60599596 0.57261322 0.40663347
4.50 1.0 1.62721177 1.67537818 0.59350053 0.38662670
5.00 1.0 1.61680140 1.73717202 0 61013216 0.37124442
5.50 1.0 1.60895981 1.78854622 0.62224623 0.35804982
6.00 1.0 1.60264144 1.83384024 0.63290617 0.34632222
6.50 1.0 1.59628453 1.87621949 0.64325250 0.33757435
7.00 1.0 1.58944625 1.91244355 0.65183500 0.32607655
/*
```

APPENDIX C

SAMPLE APPLICATIONS

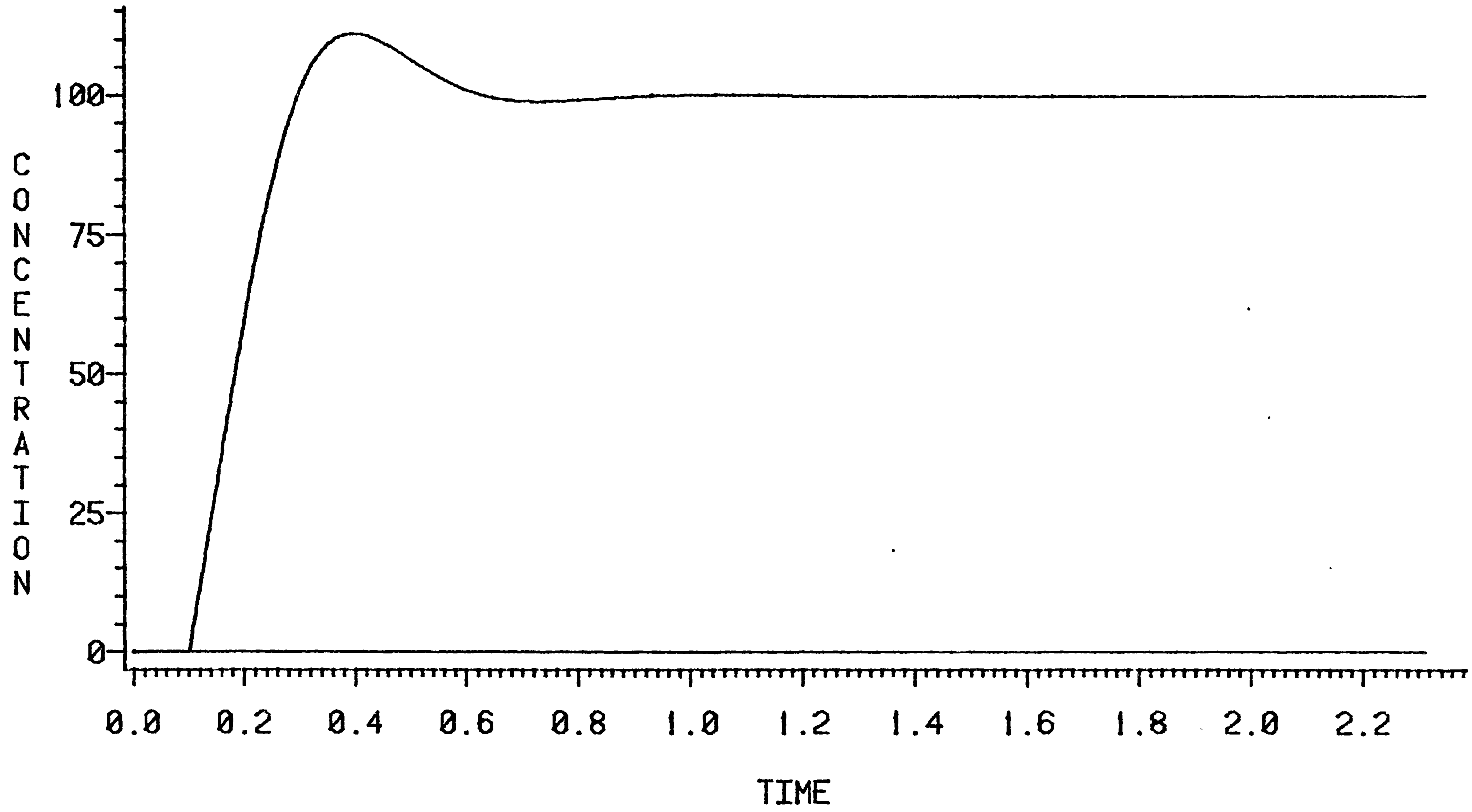


Figure 35. Concentration Variables, PI Controller, θ_d/τ_1 = 0.1, r = 0.4

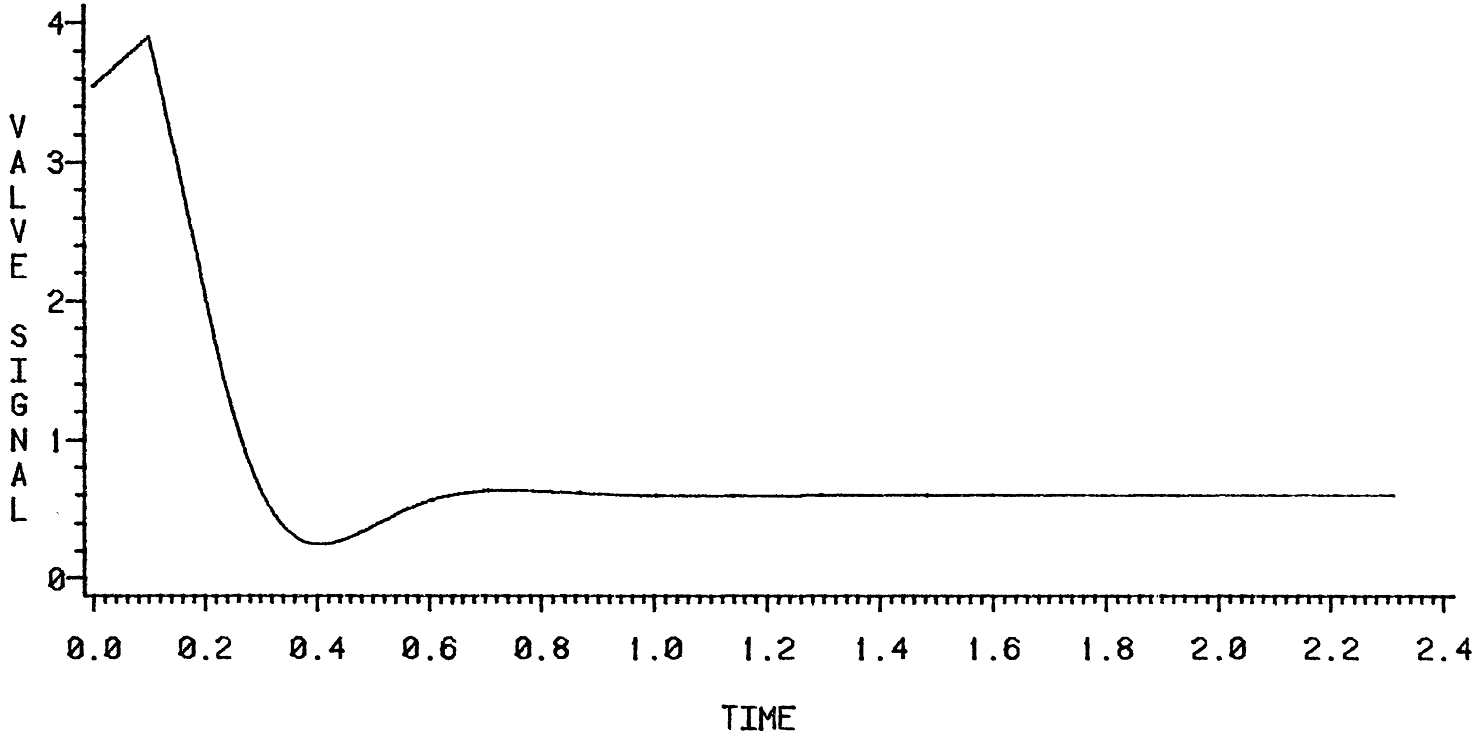


Figure 36. Valve Signal, PI Controller, θ_d/τ_1 = 0.1, r = 0.4

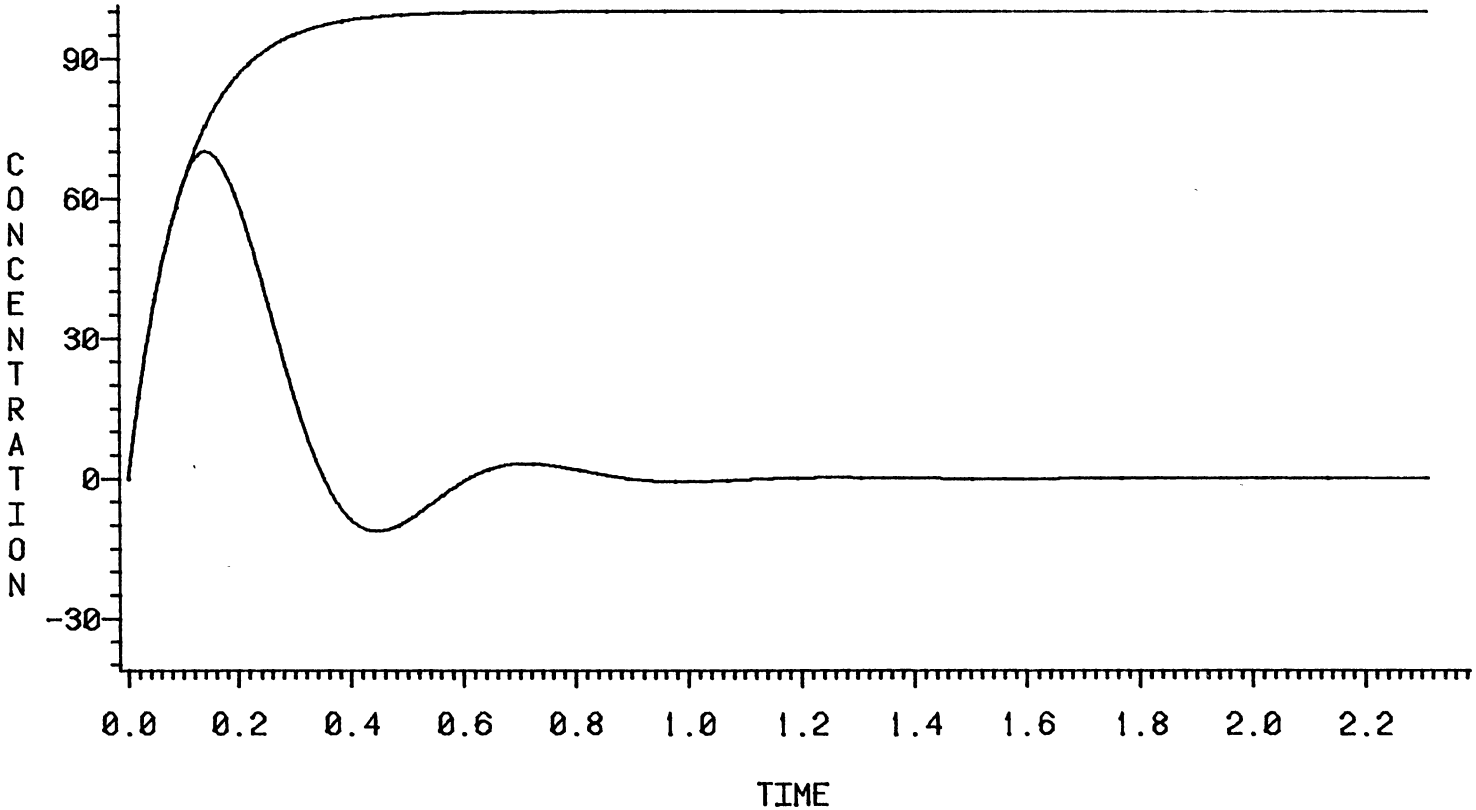


Figure 37. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.1$

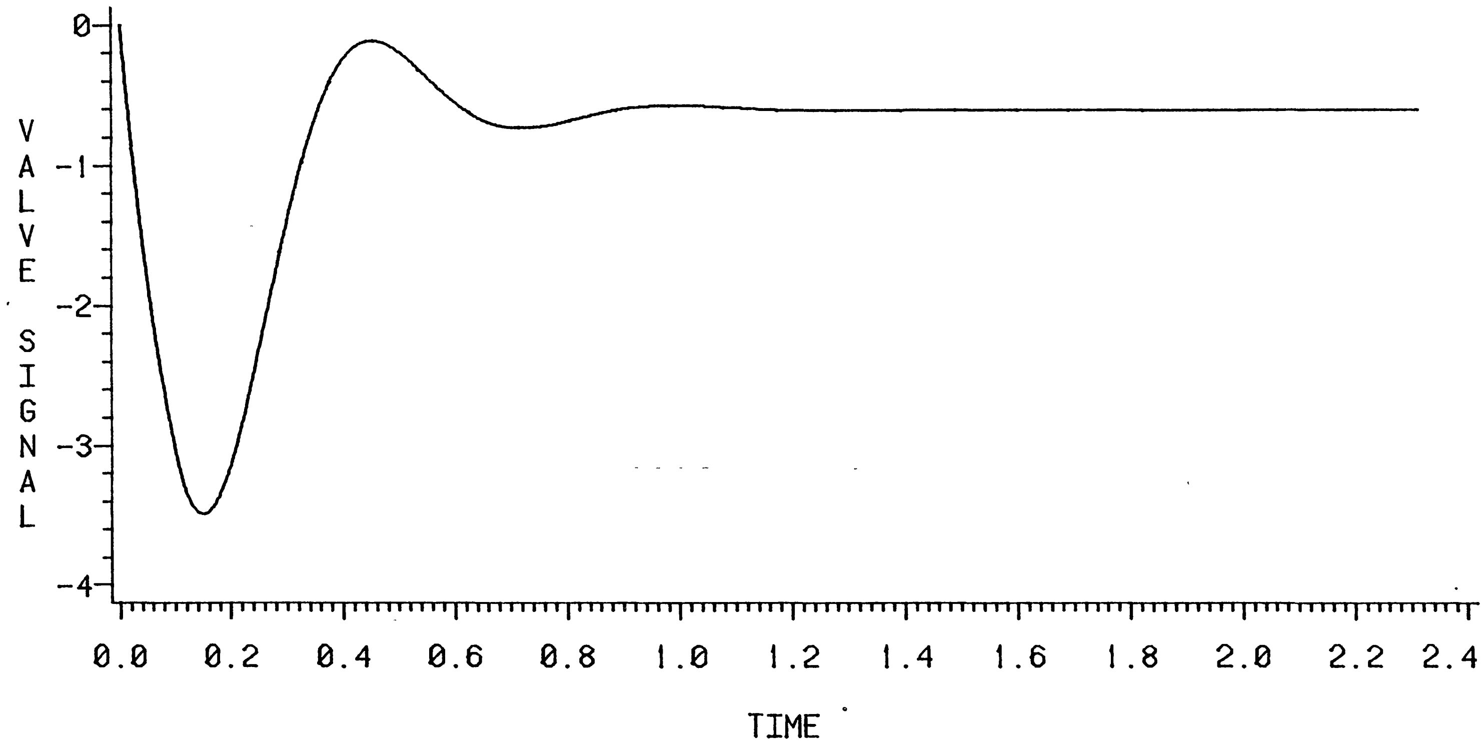


Figure 38. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.1$

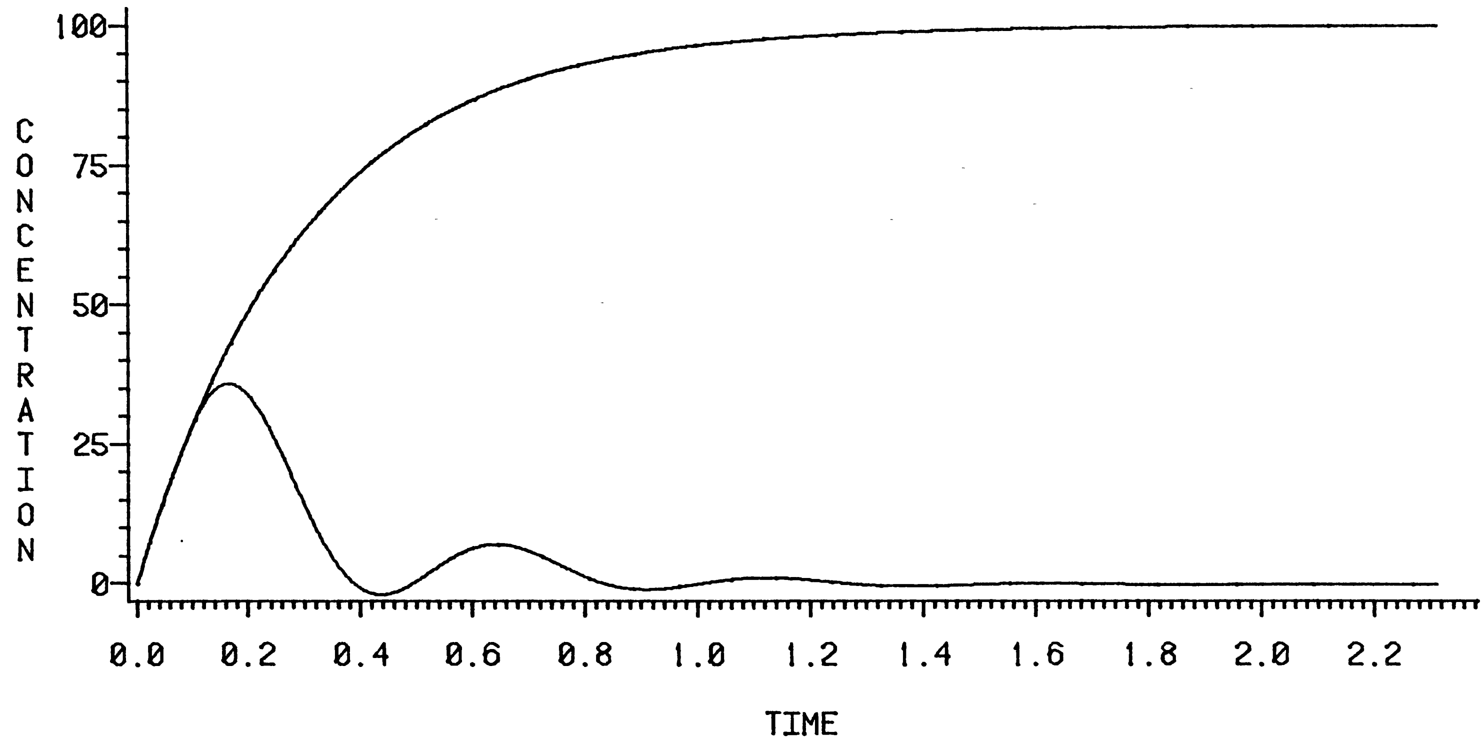


Figure 39. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.3$

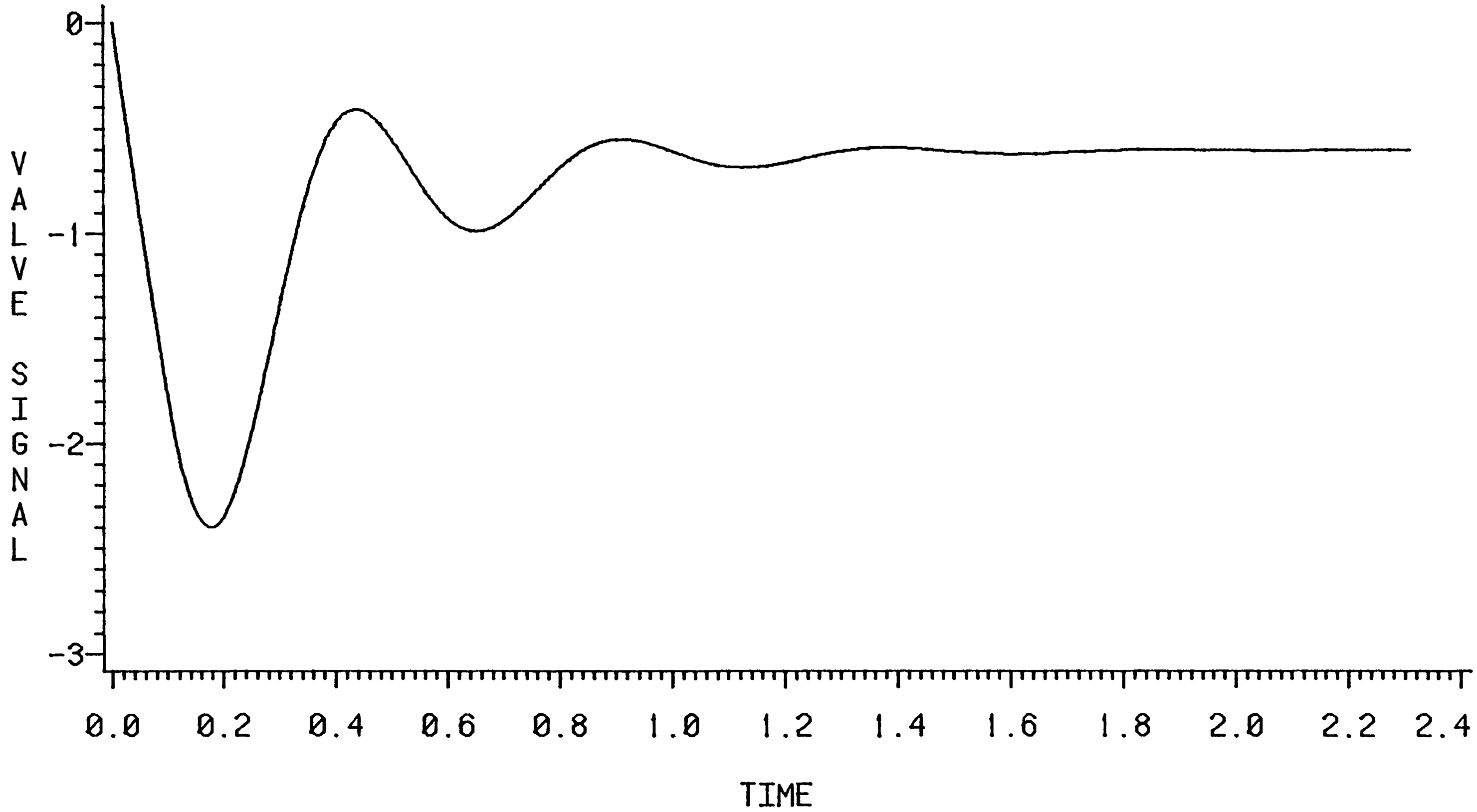


Figure 40. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.3$

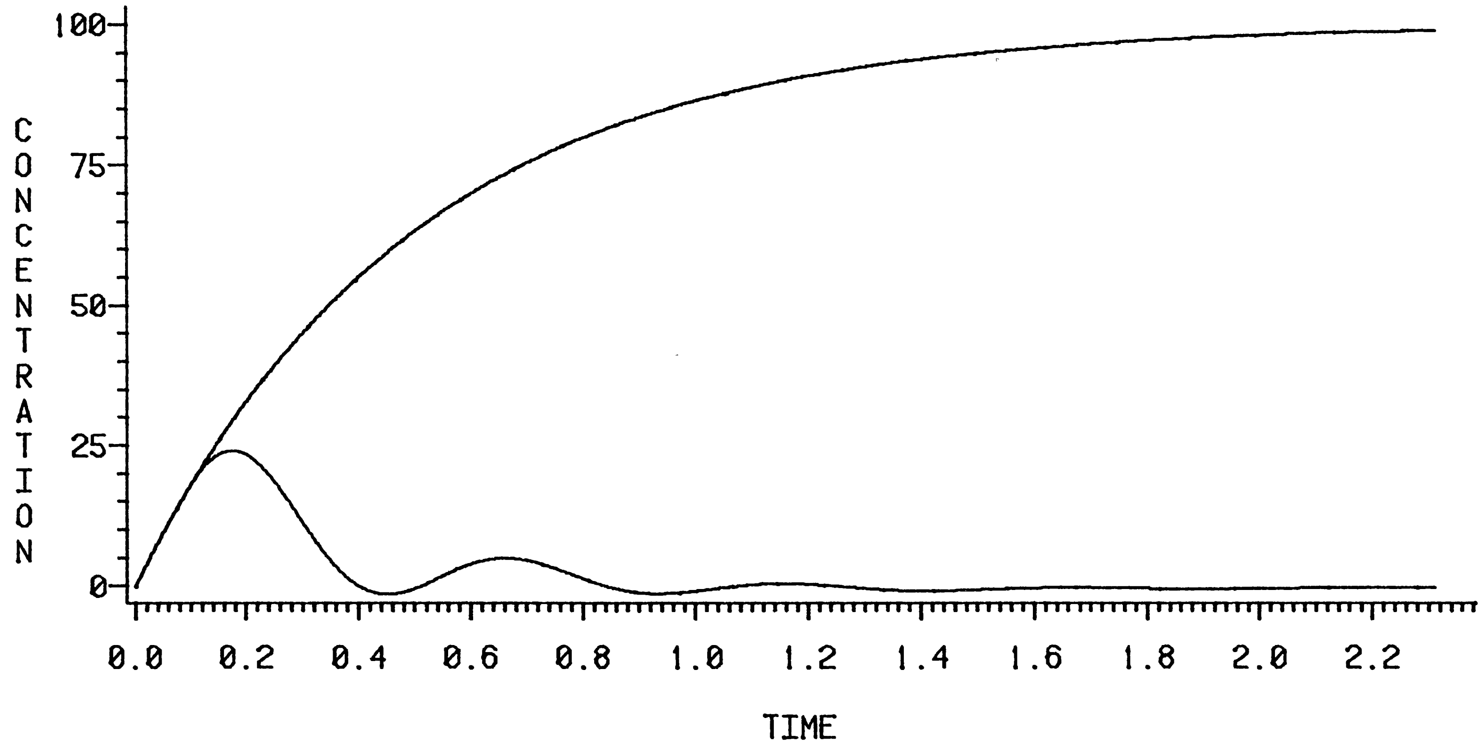


Figure 41. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.5$

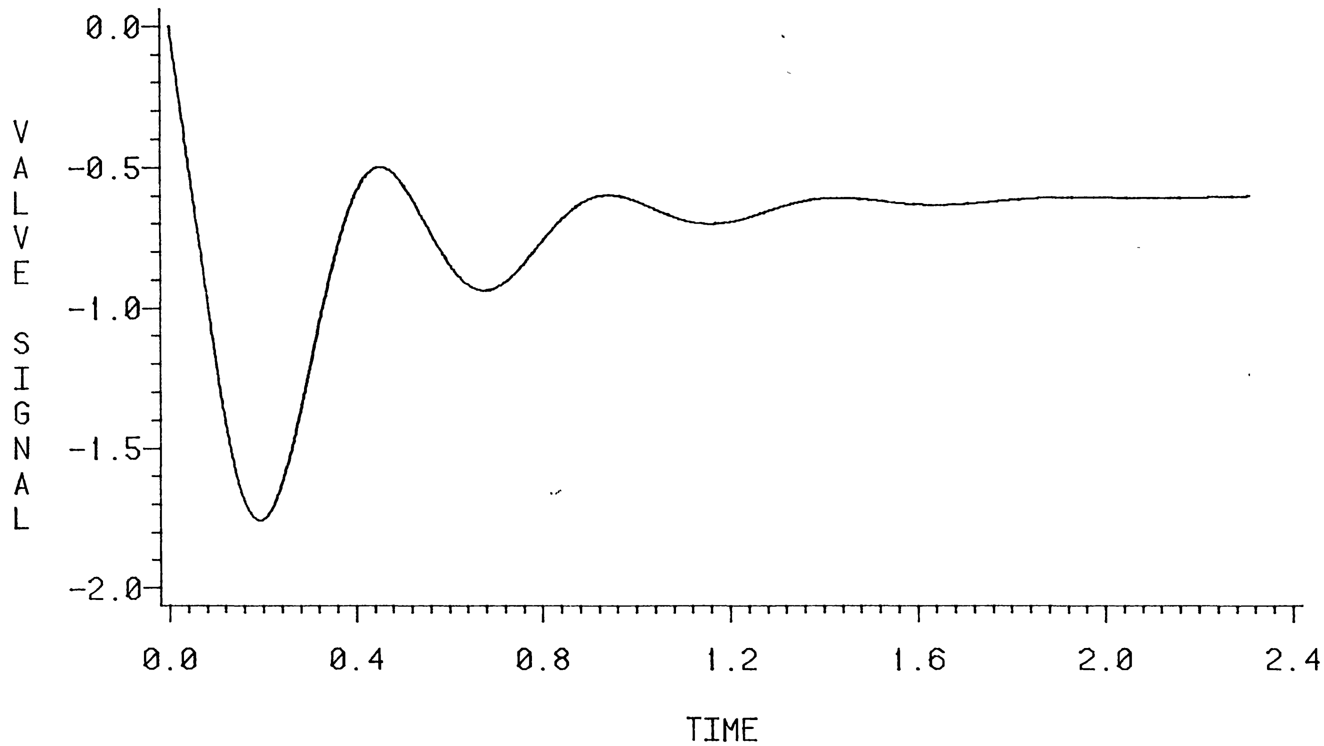


Figure 42. Valve Signal, PI Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 0.5

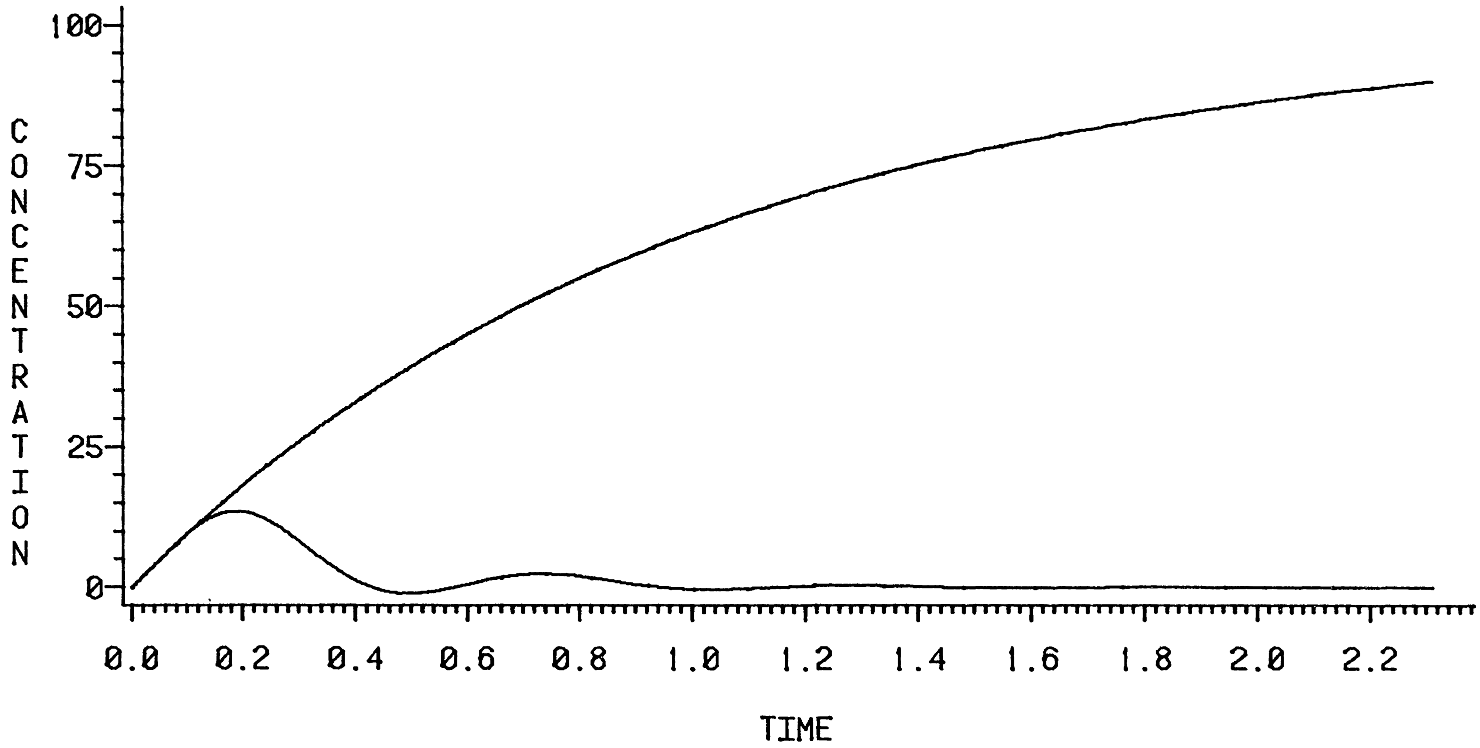


Figure 43. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 1.0$

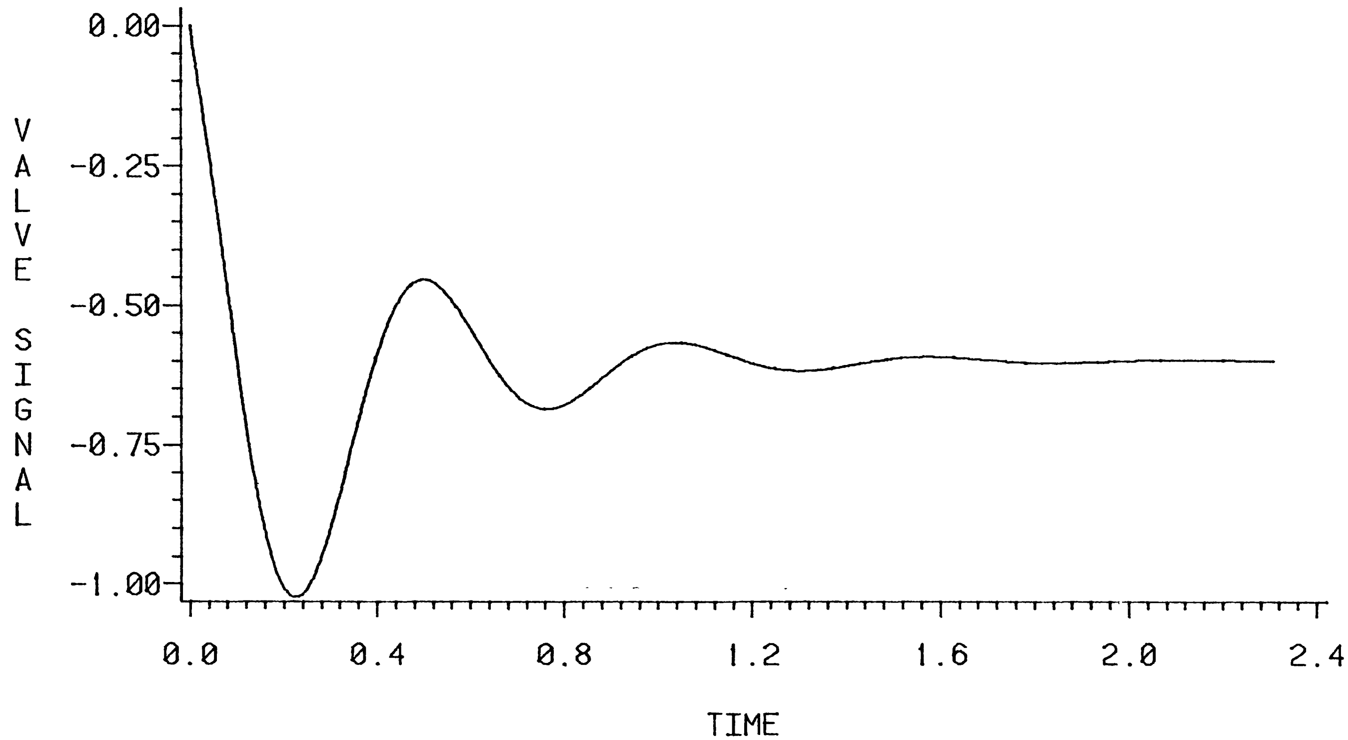


Figure 44. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 1.0$

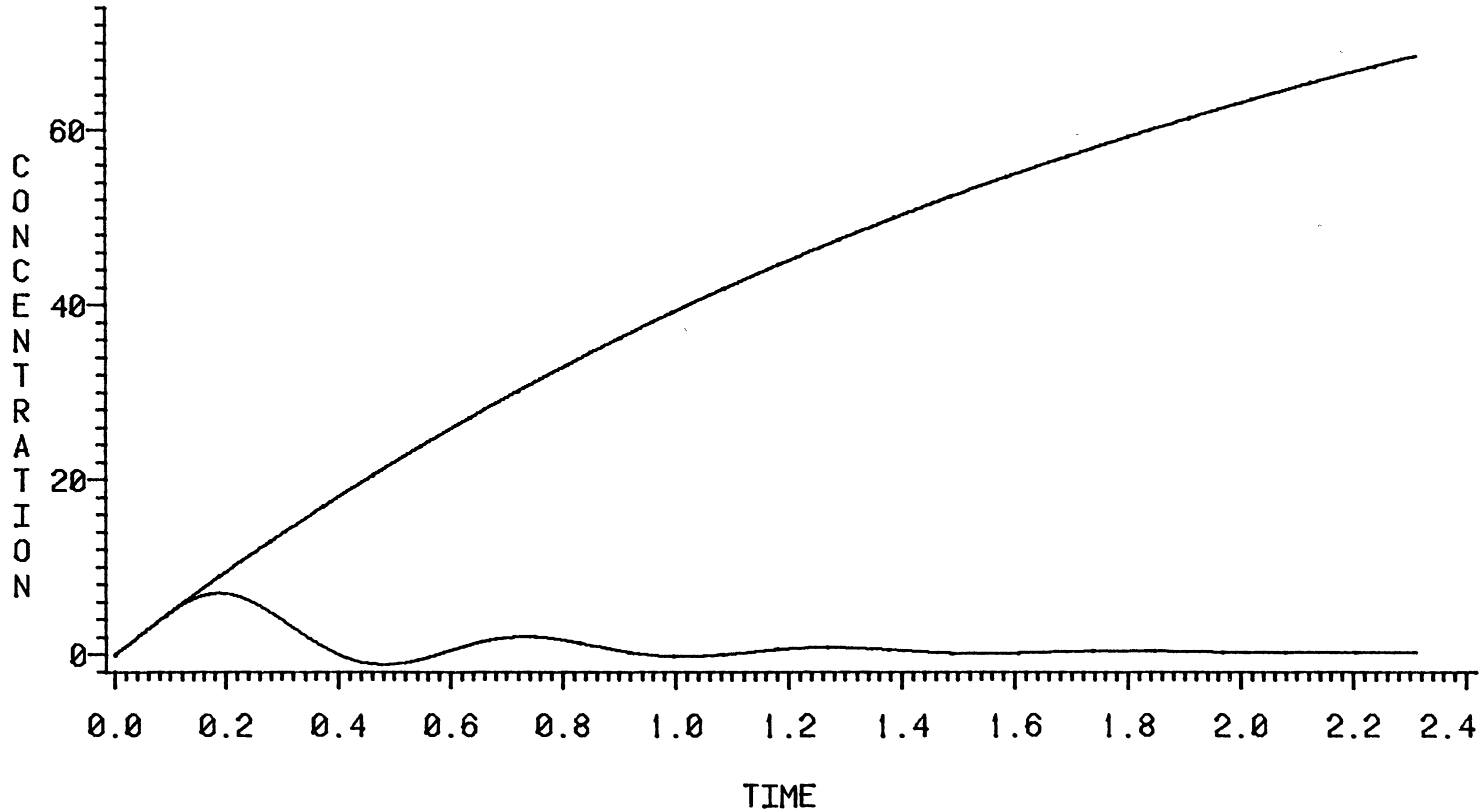


Figure 45. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 2.0$

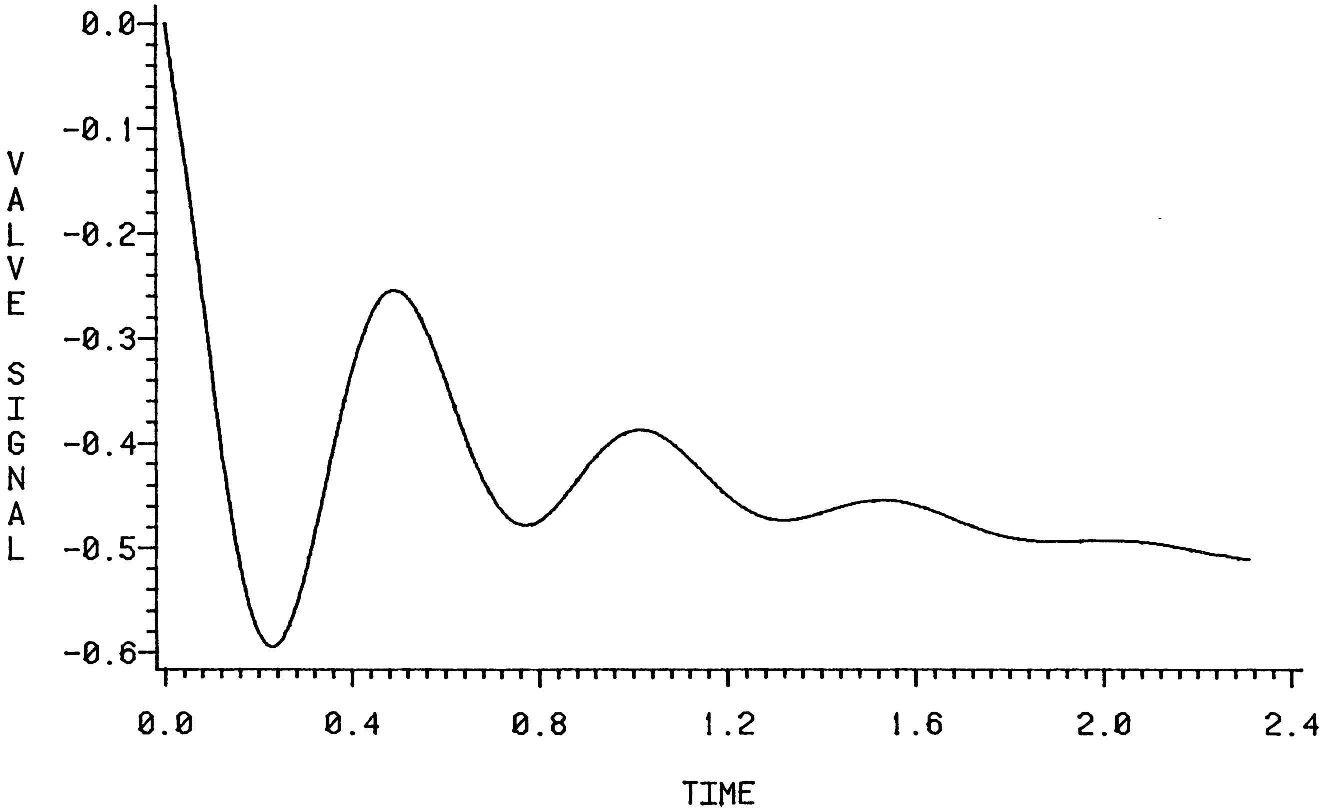


Figure 46. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 2.0$

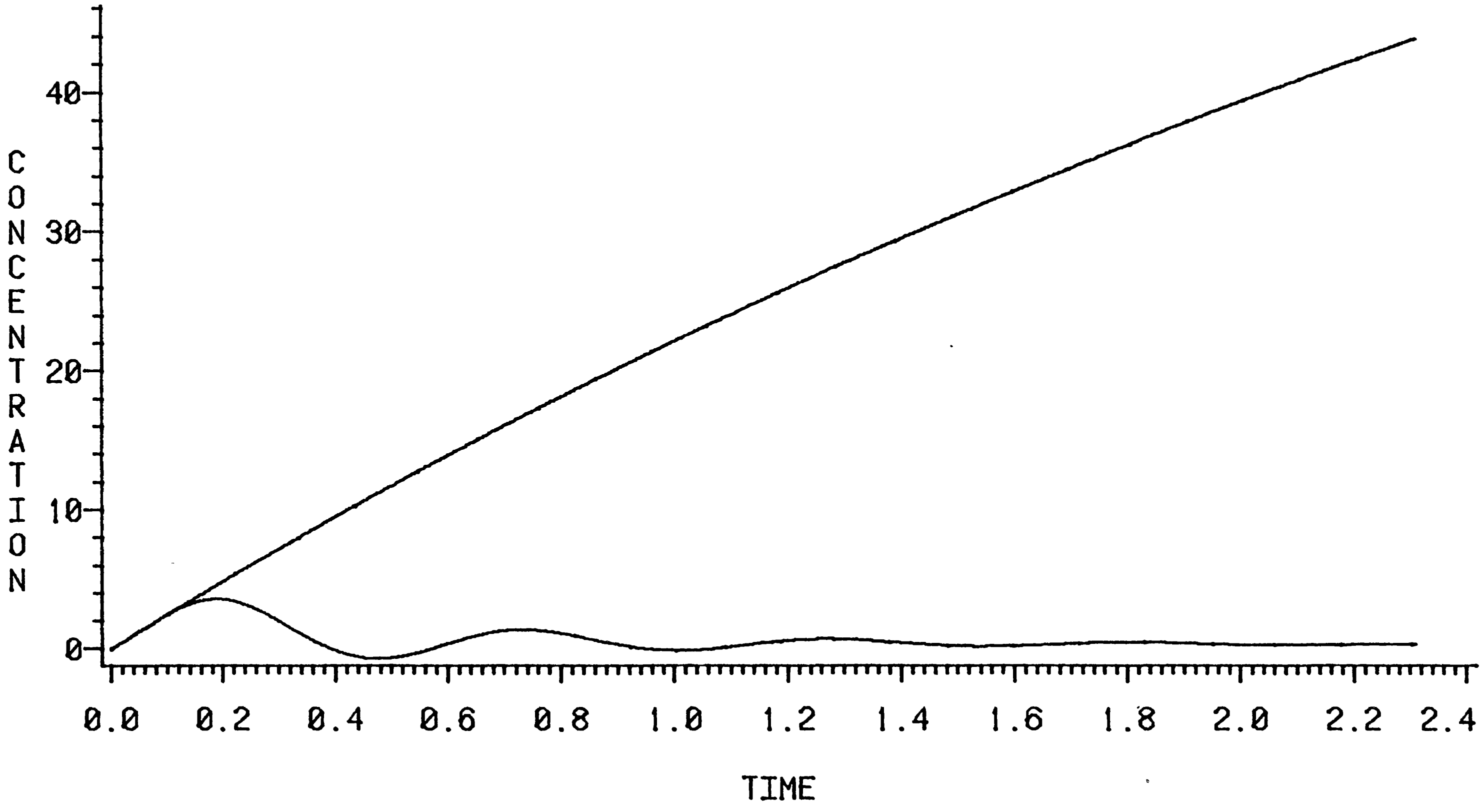


Figure 47. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 4.0$

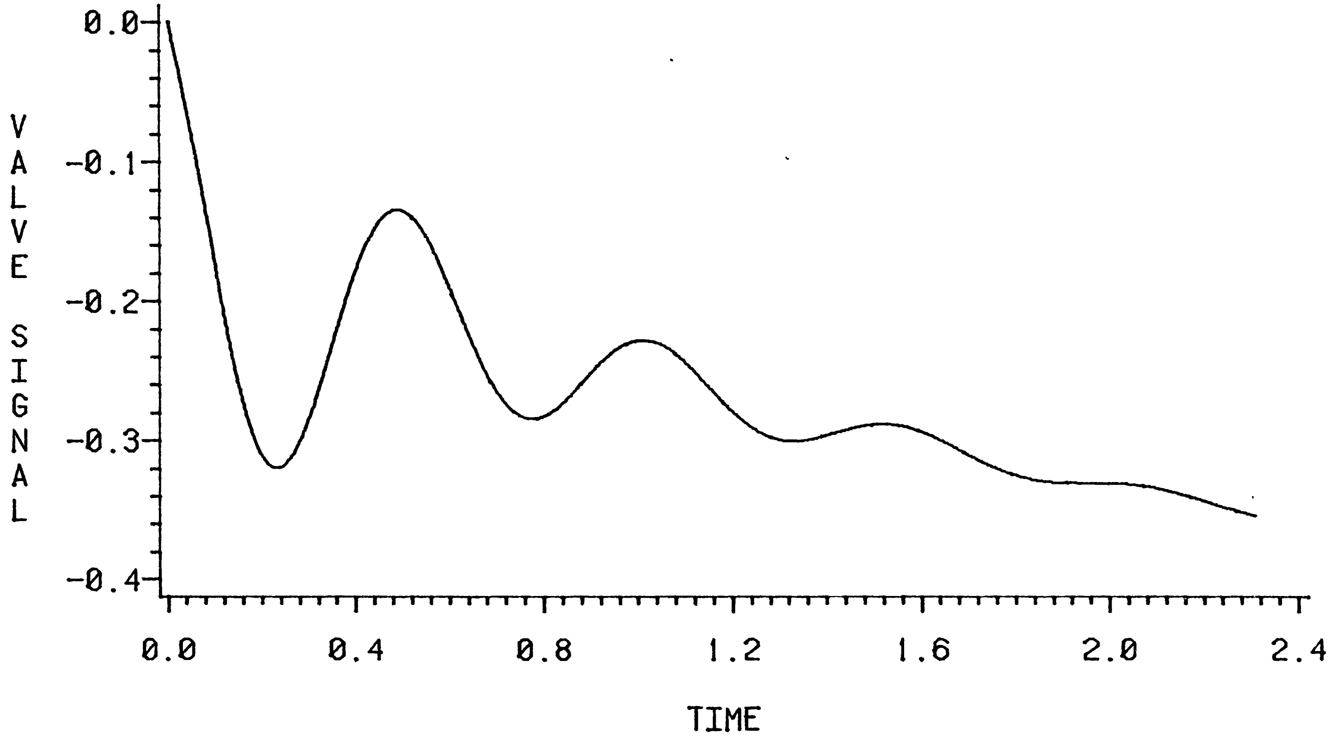


Figure 48. Valve Signal, PI Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 4.0

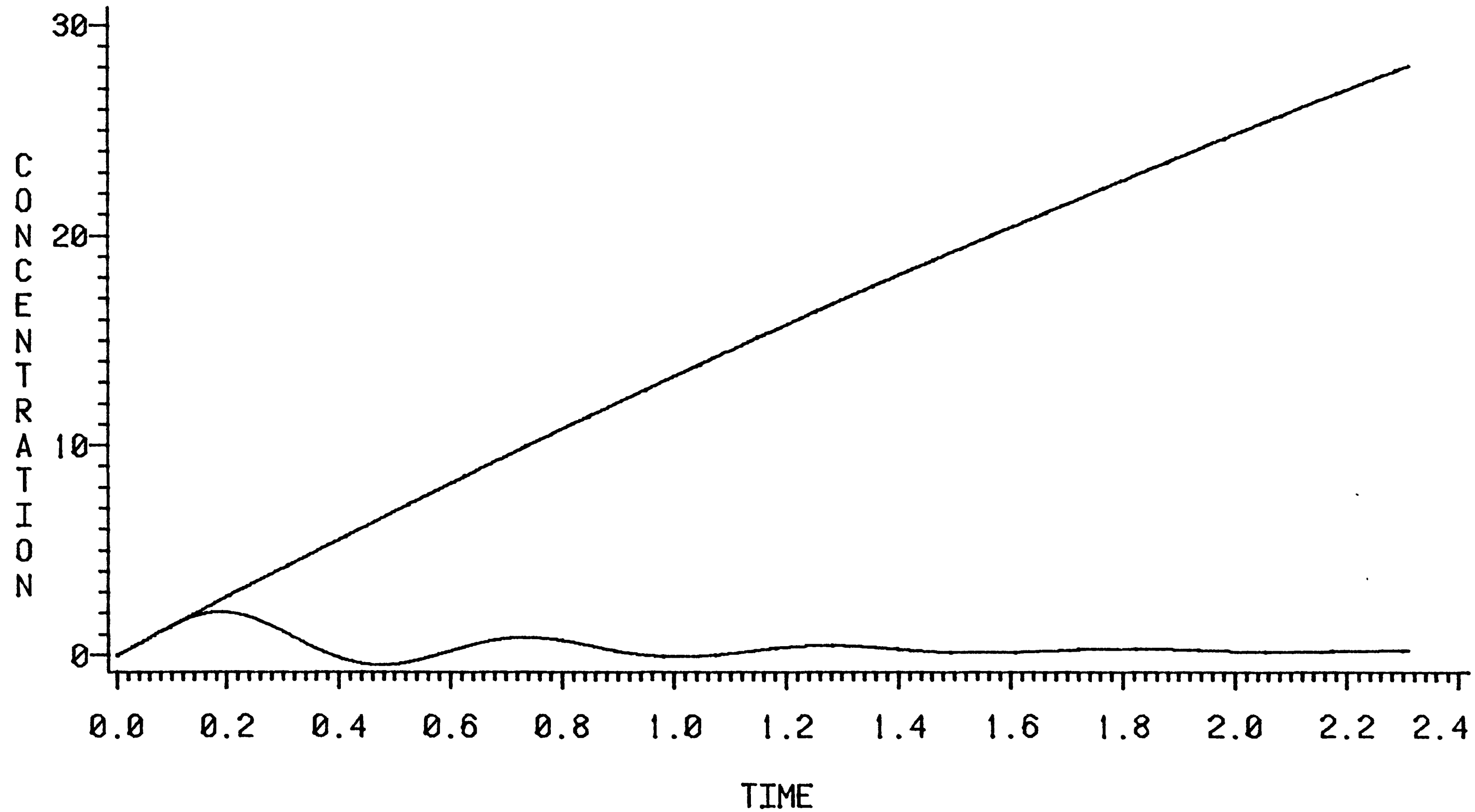


Figure 49. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 7.0$

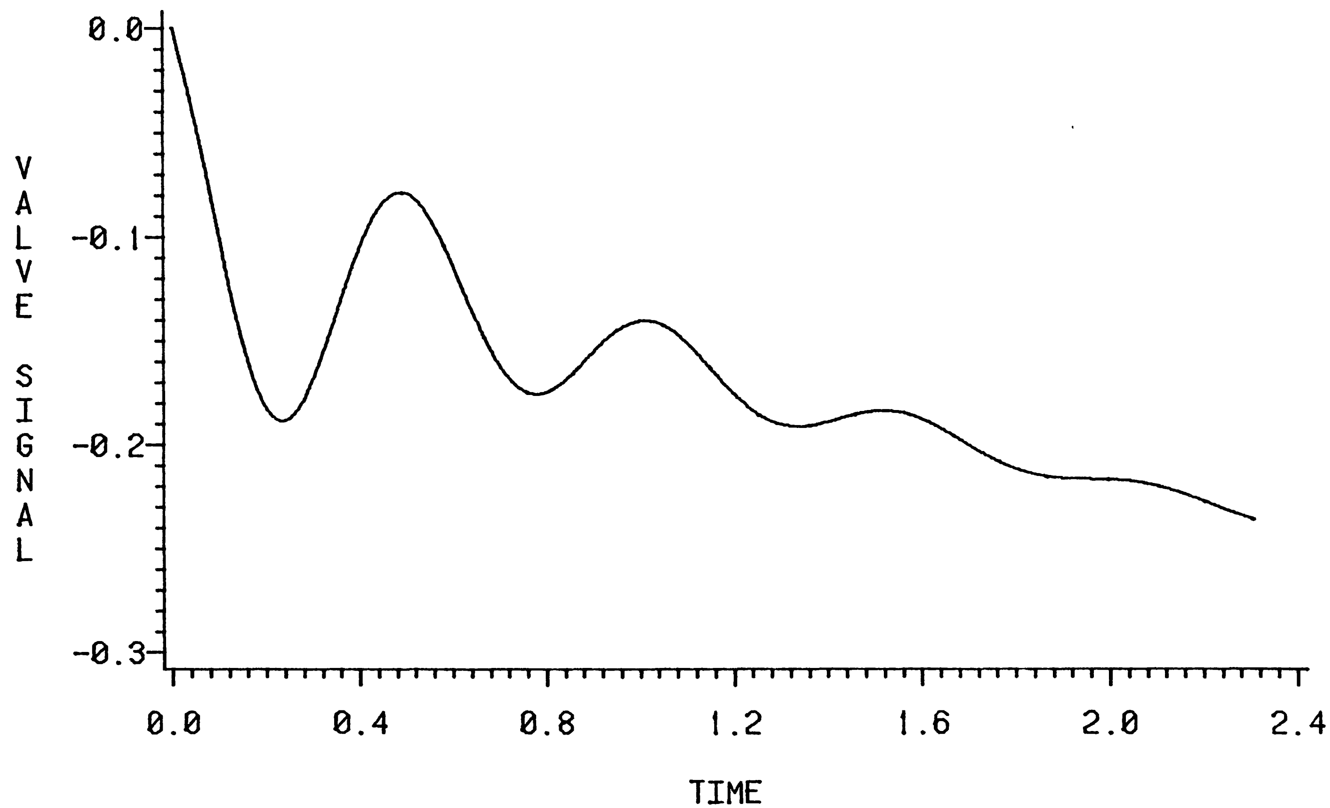


Figure 50. Valve Signal, PI Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 7.0

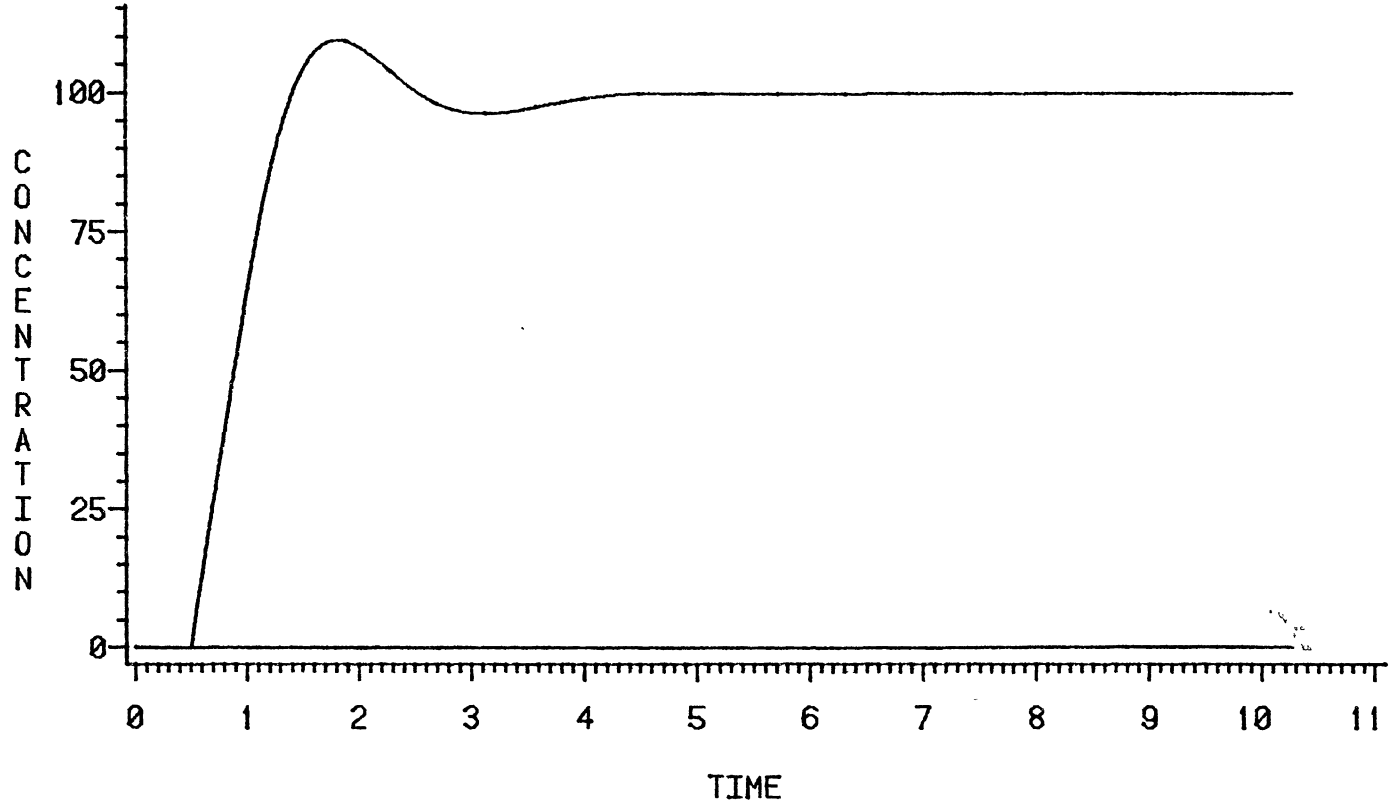


Figure 51. Concentration Variables, PI Controller, θ_d/τ_1 = 0.5, r = 0.4

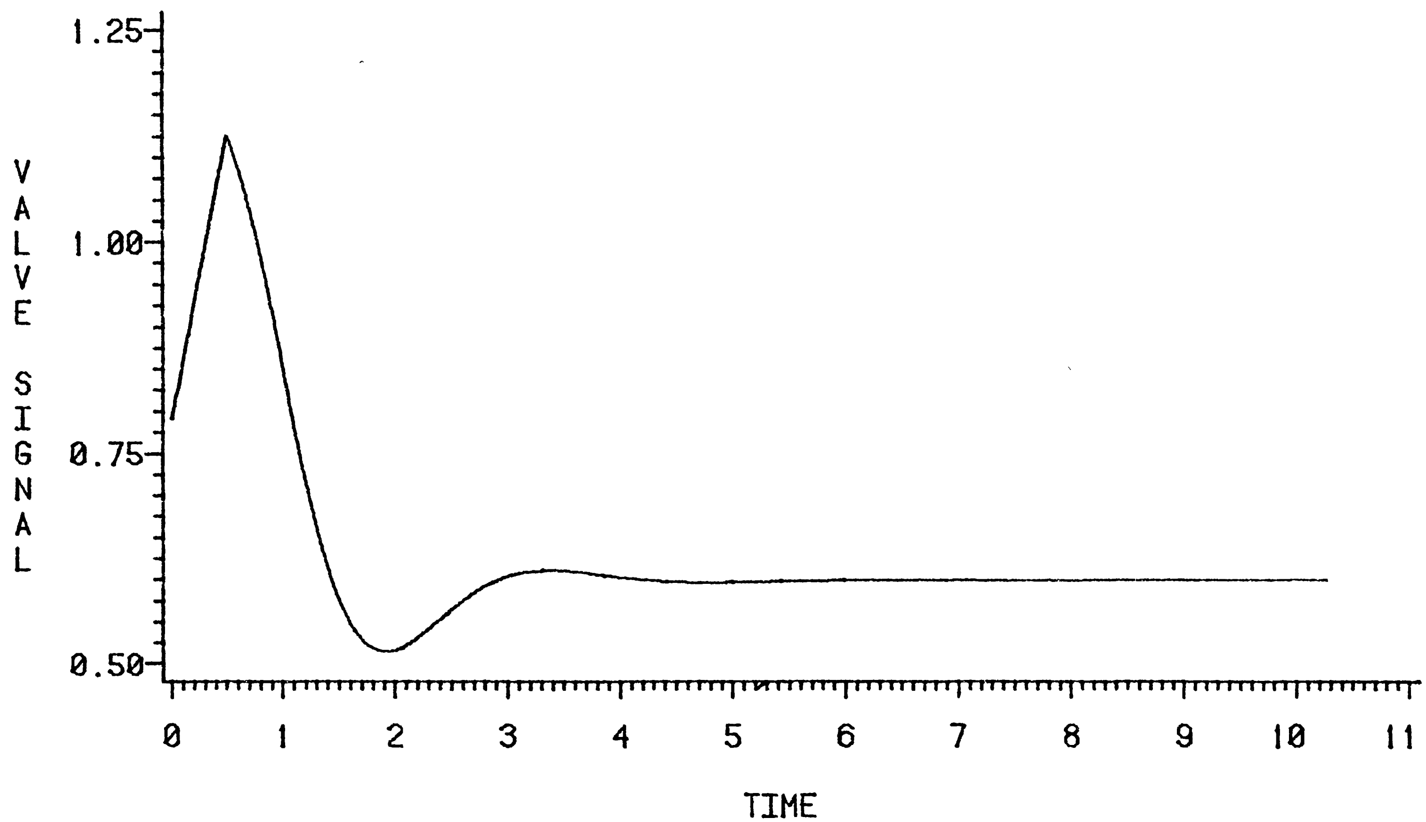


Figure 52. Valve Signal, PI Controller, θ_d/τ_1 = 0.5, r = 0.4

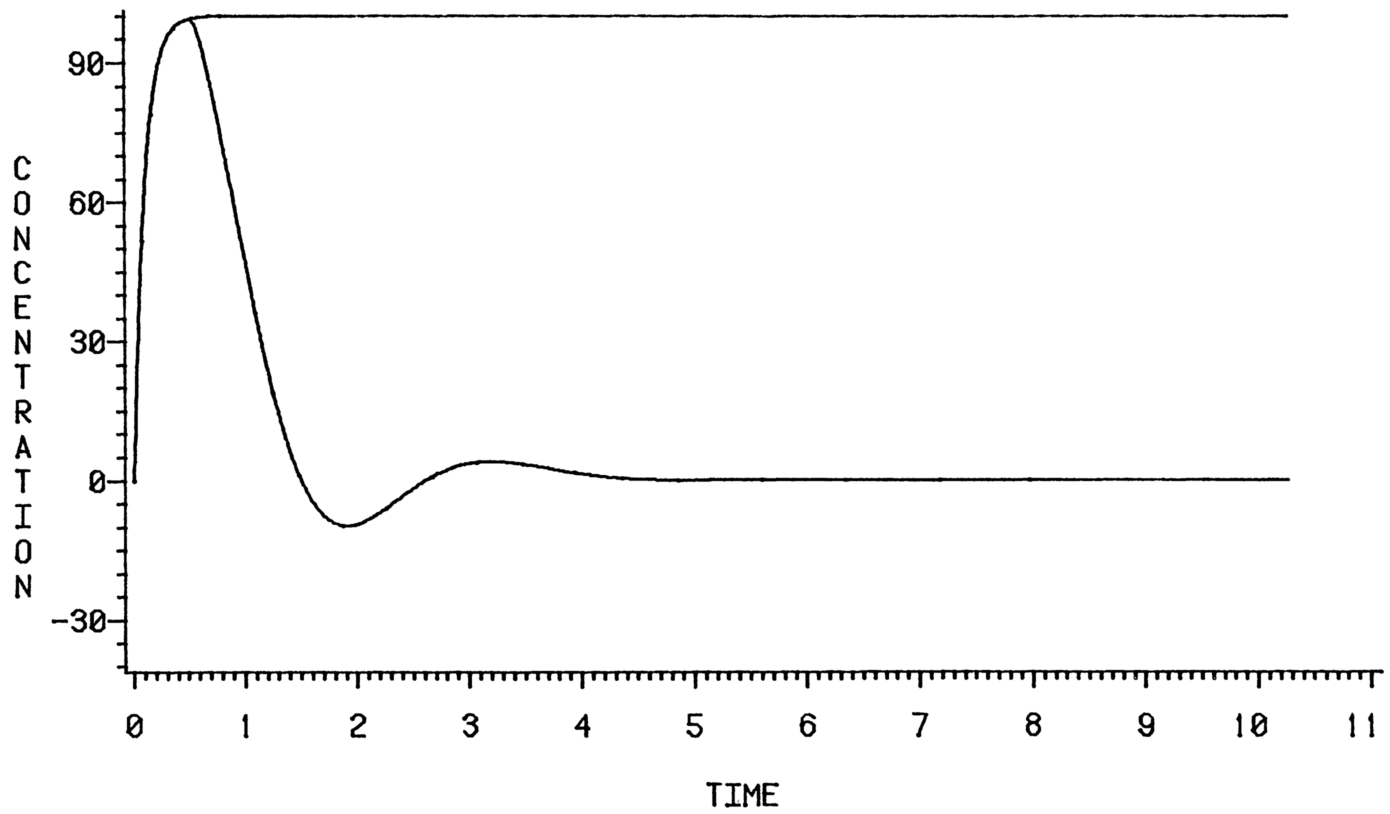


Figure 53. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.1$

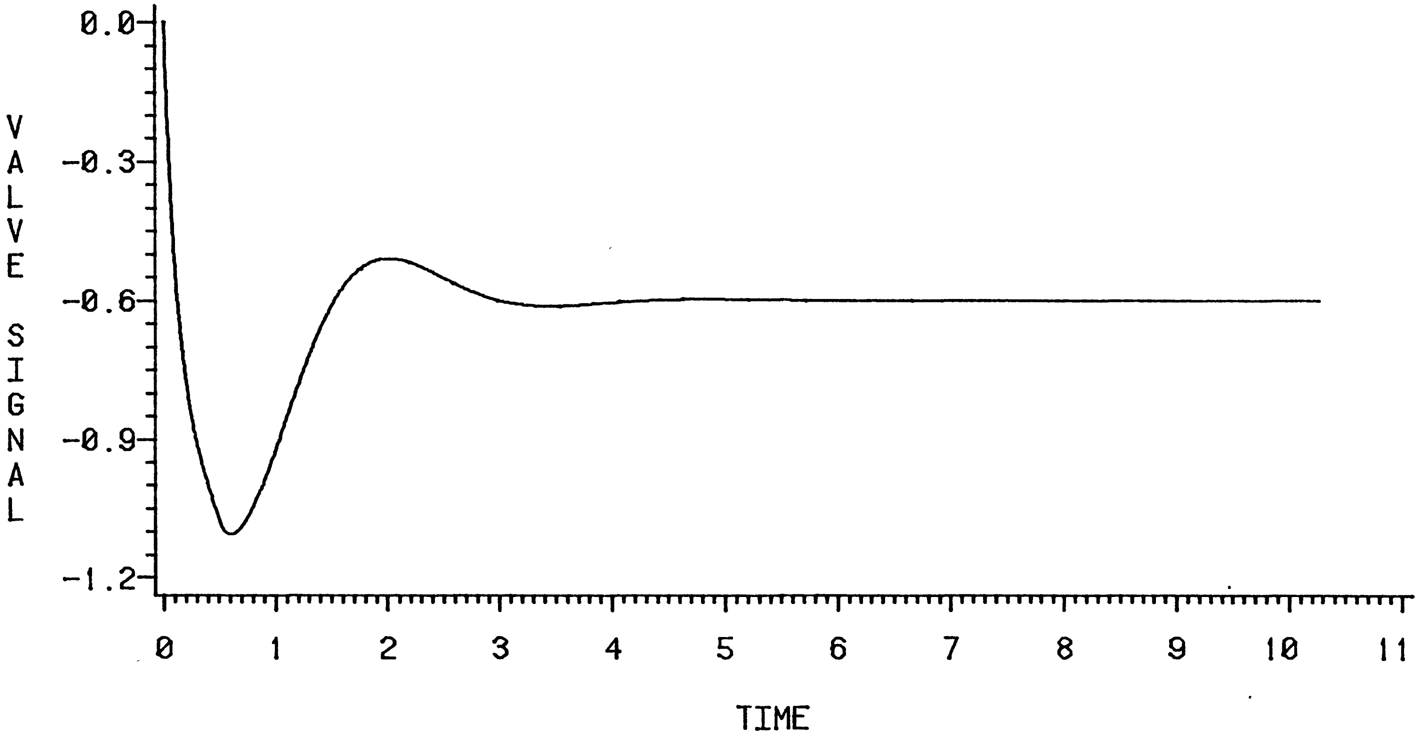


Figure 54. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.1$

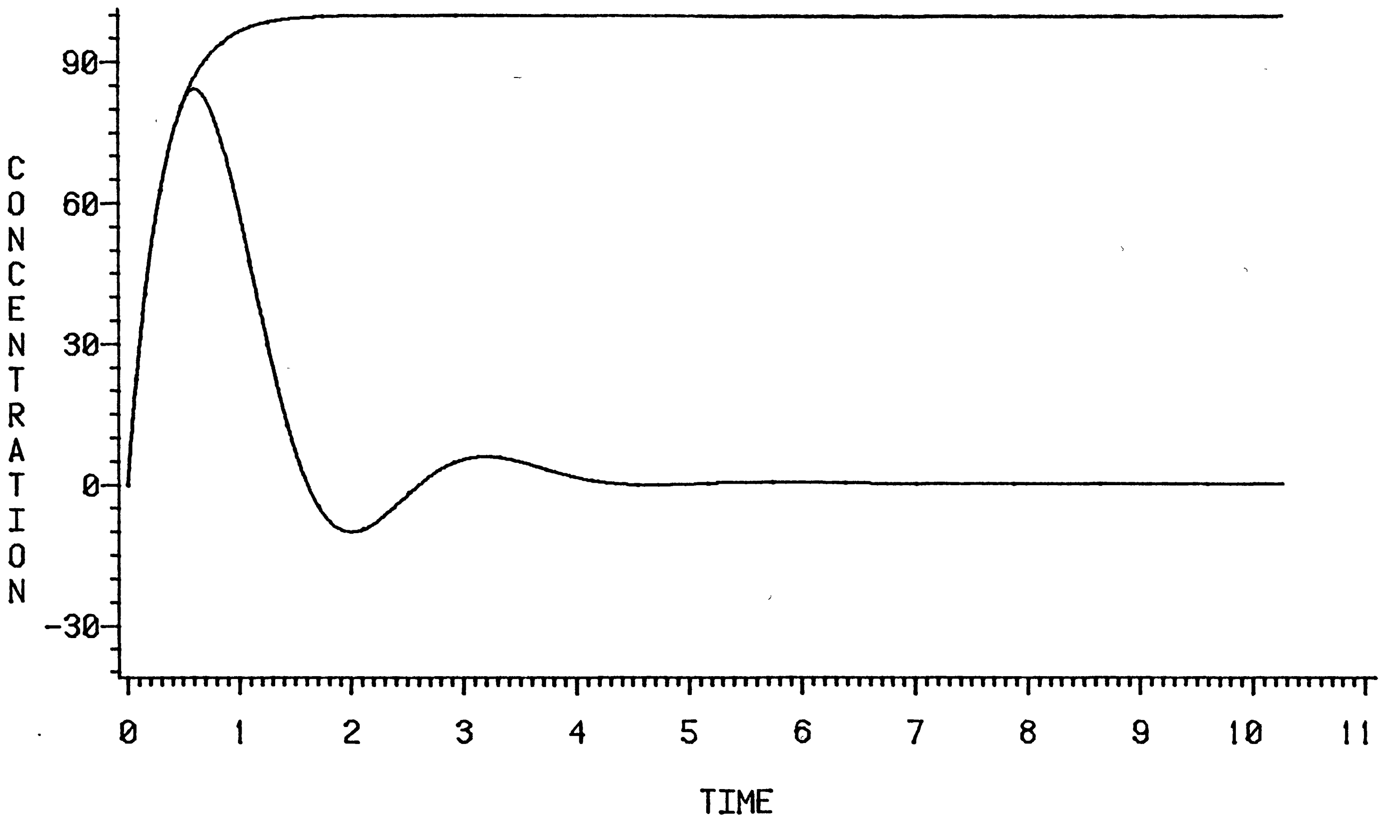


Figure 55. Concentration Variables, PI Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 0.3

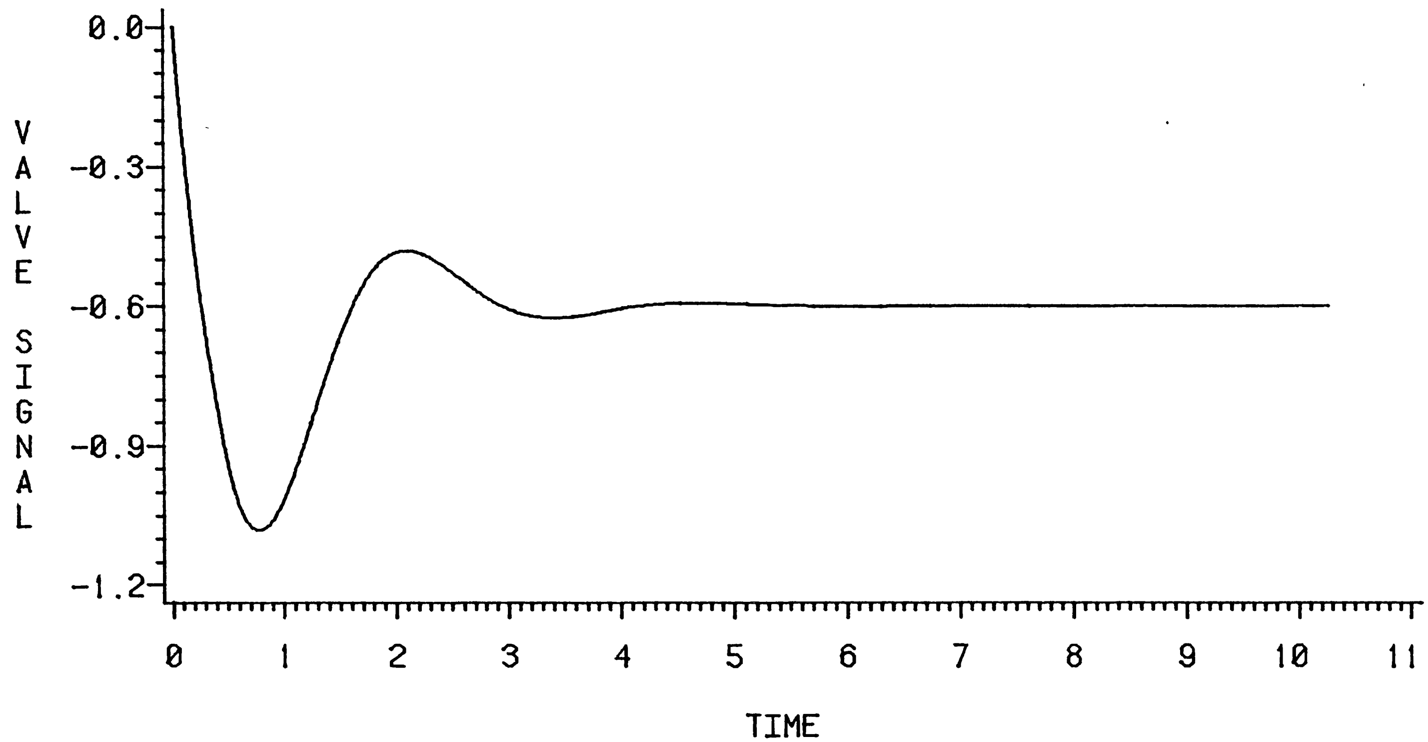


Figure 56. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.3$

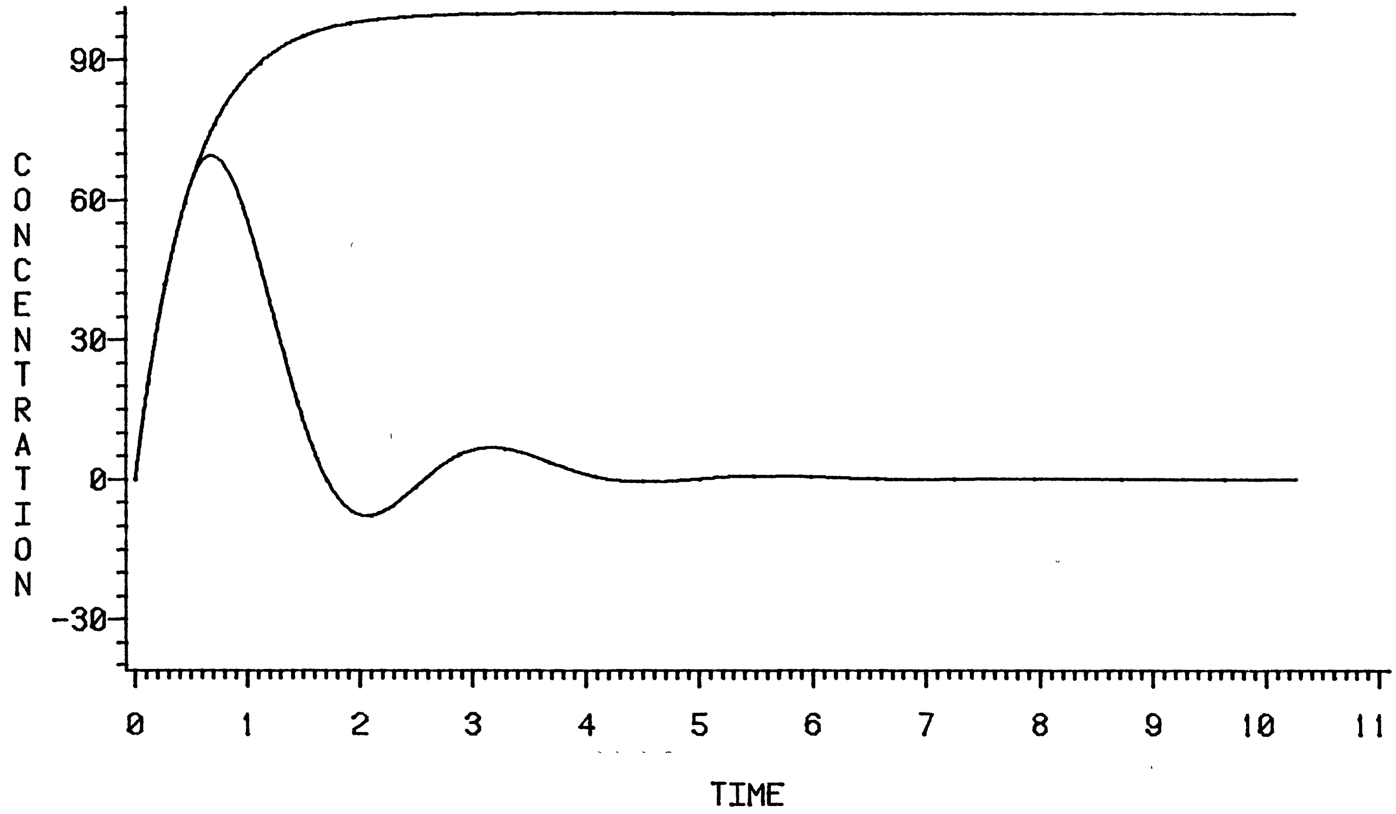


Figure 57. Concentration Variables, PI Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 0.5

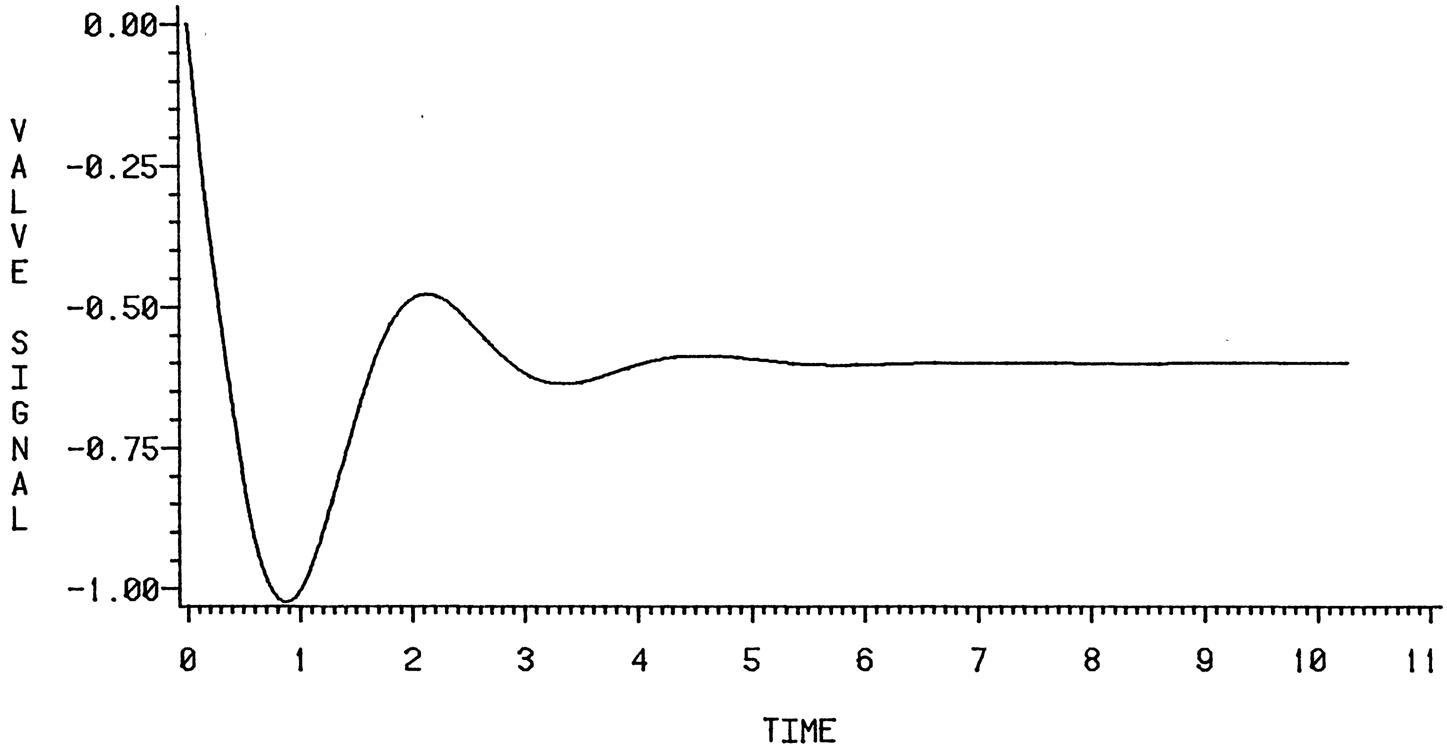


Figure 58. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.5$

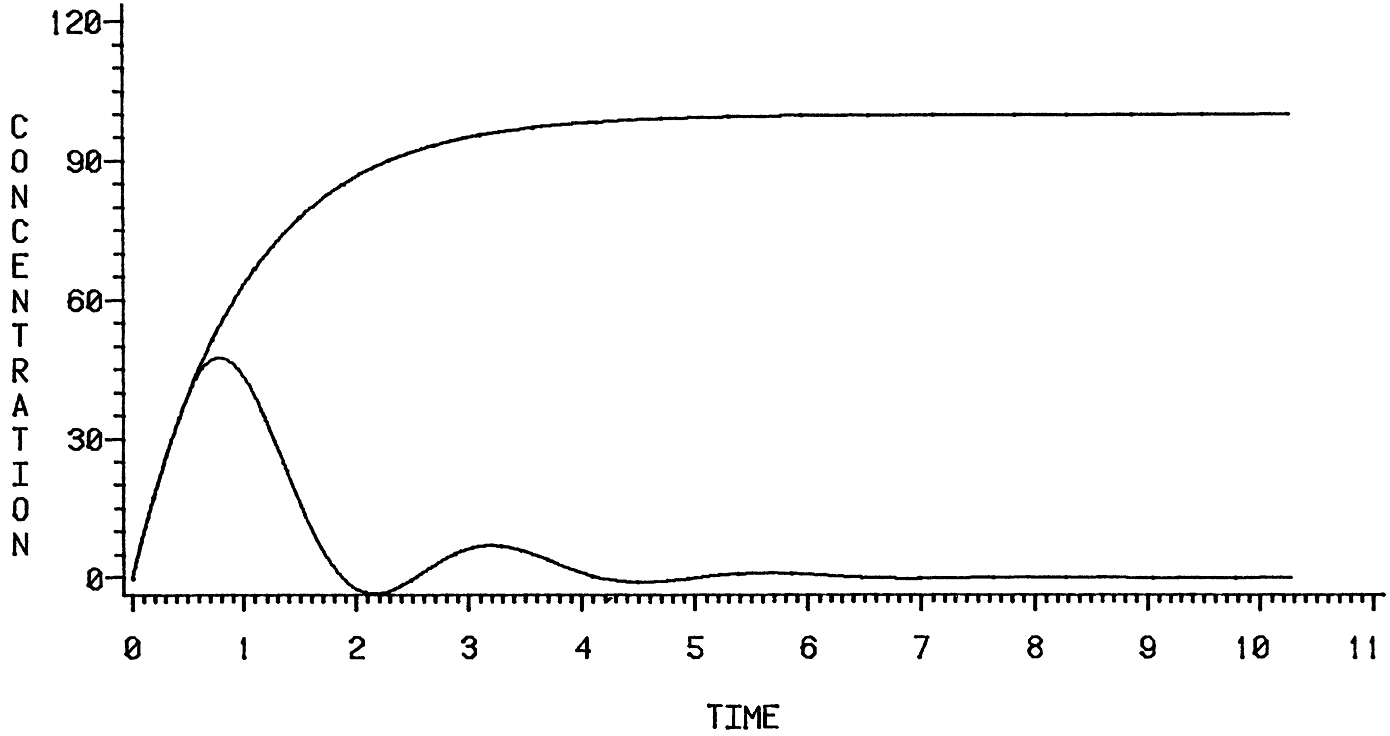


Figure 59. Concentration Variables, PI Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 1.0

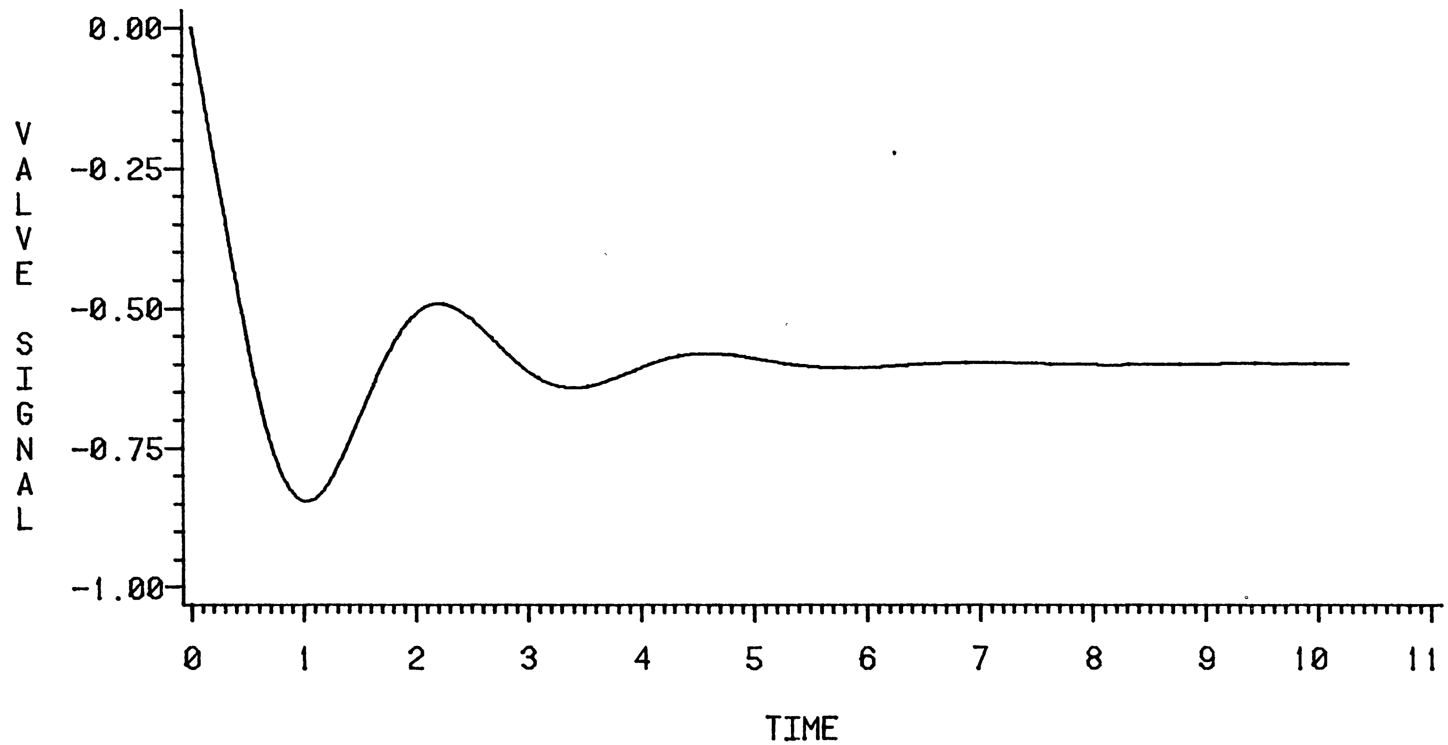


Figure 60. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 1.0$

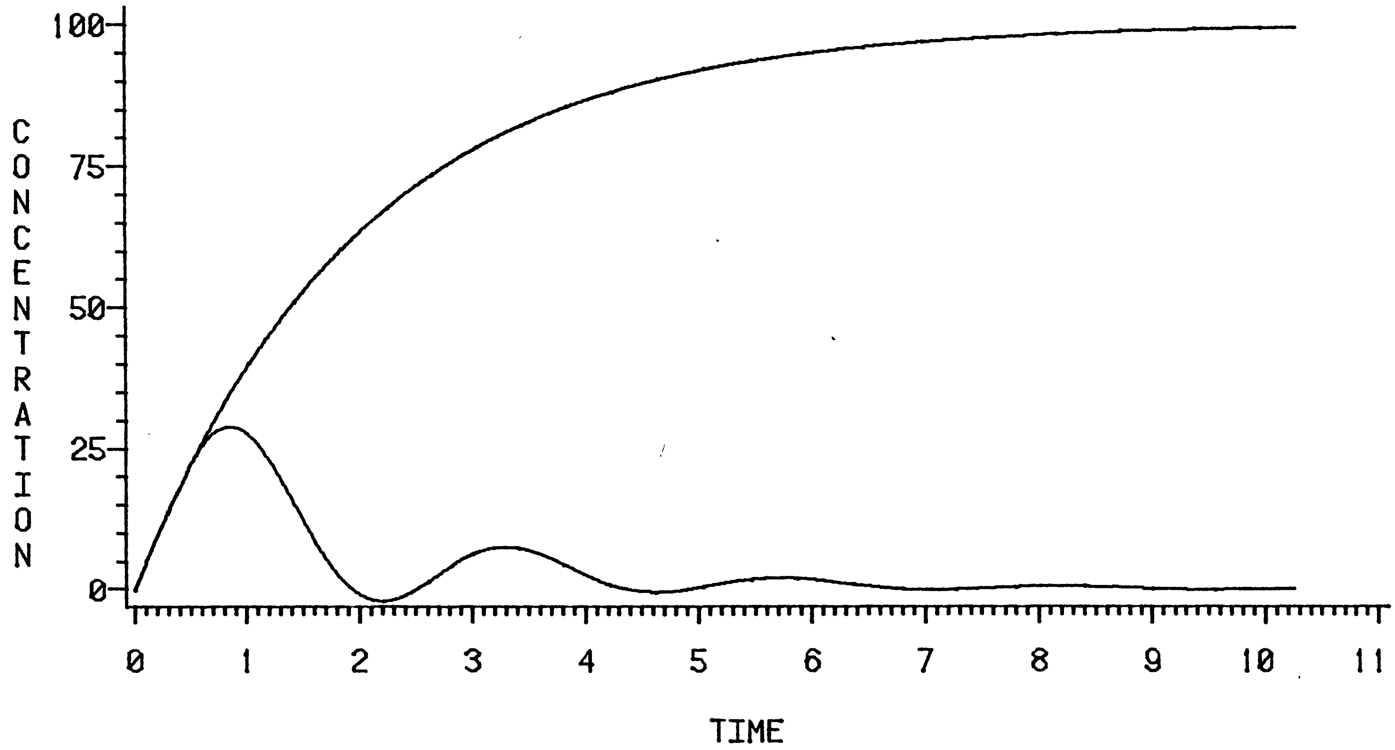


Figure 61. Concentration Variables, PI Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 2.0

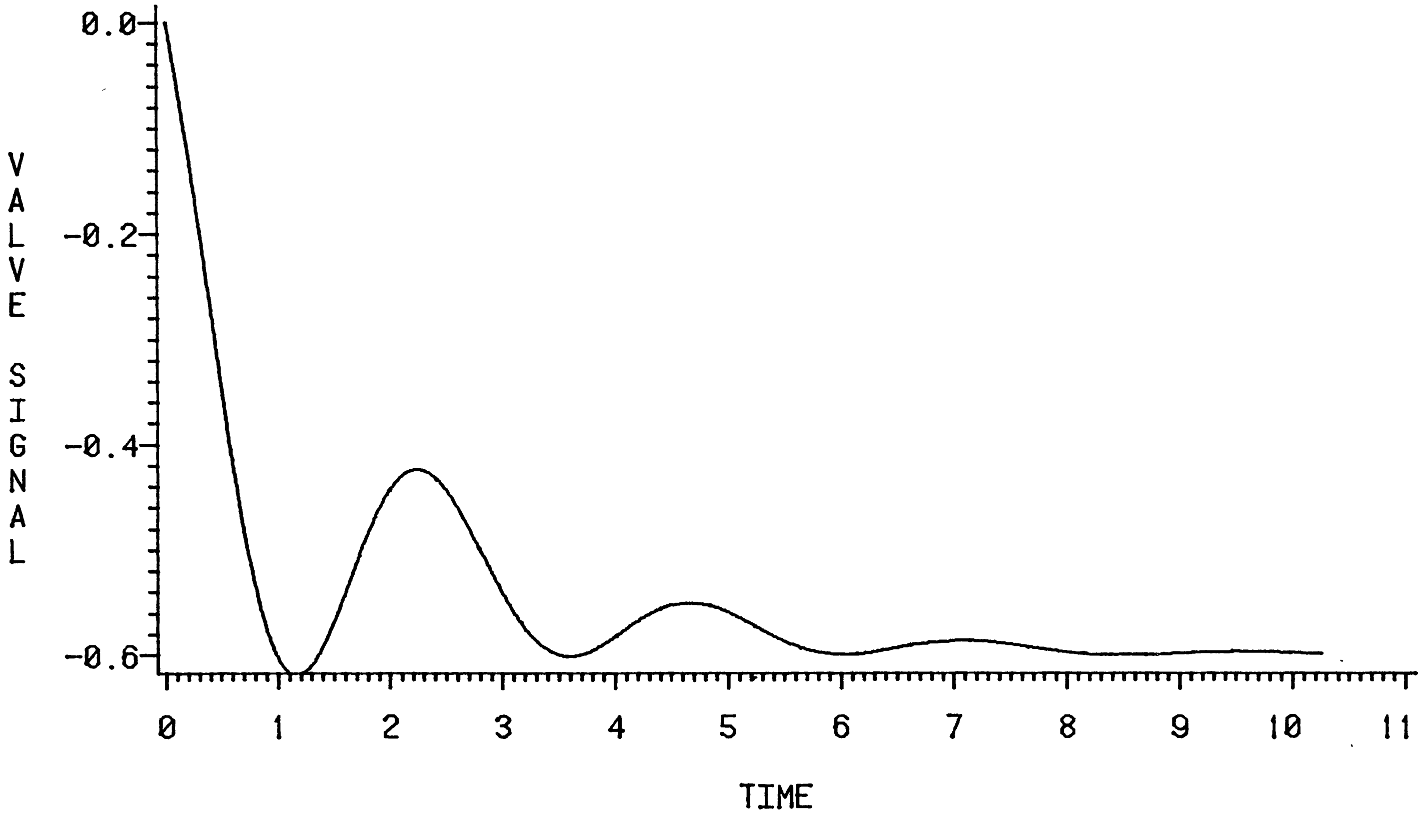


Figure 62. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 2.0$

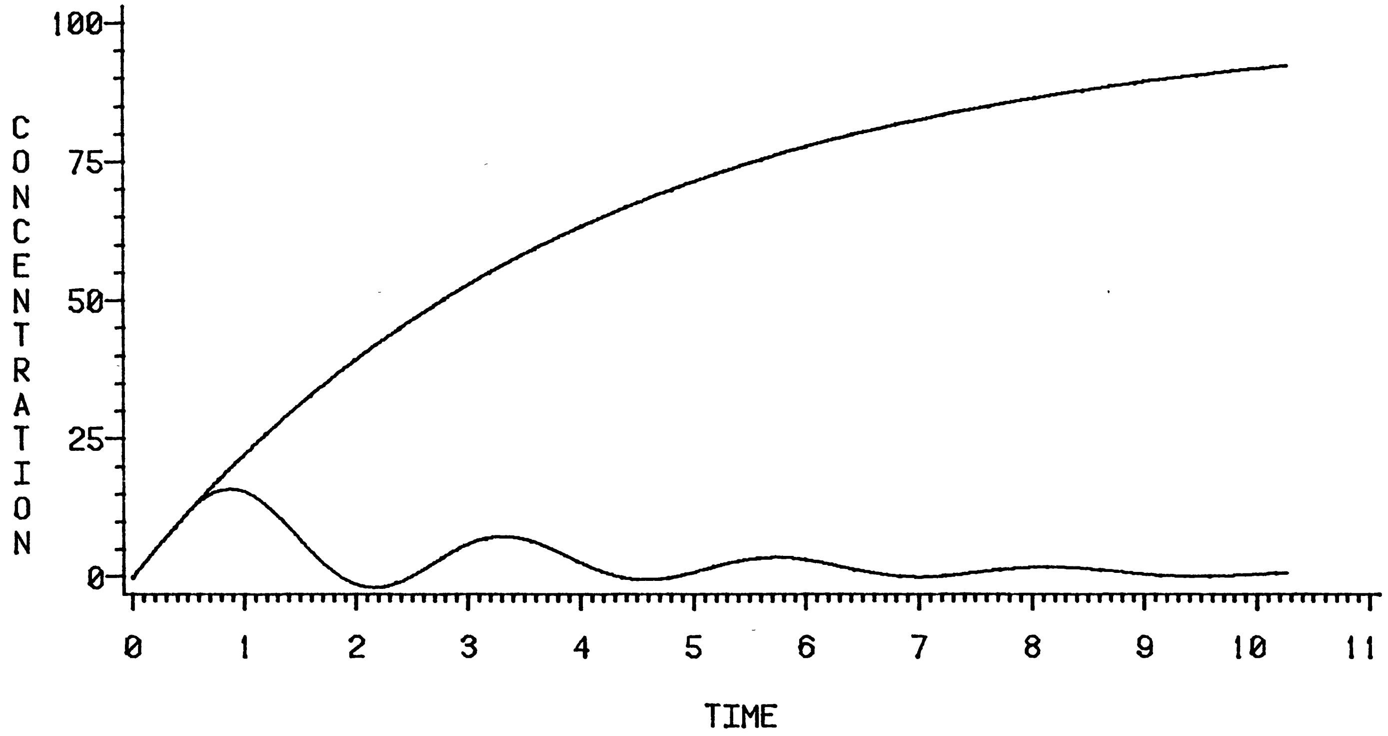


Figure 63. Concentration Variables, PI Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 4.0

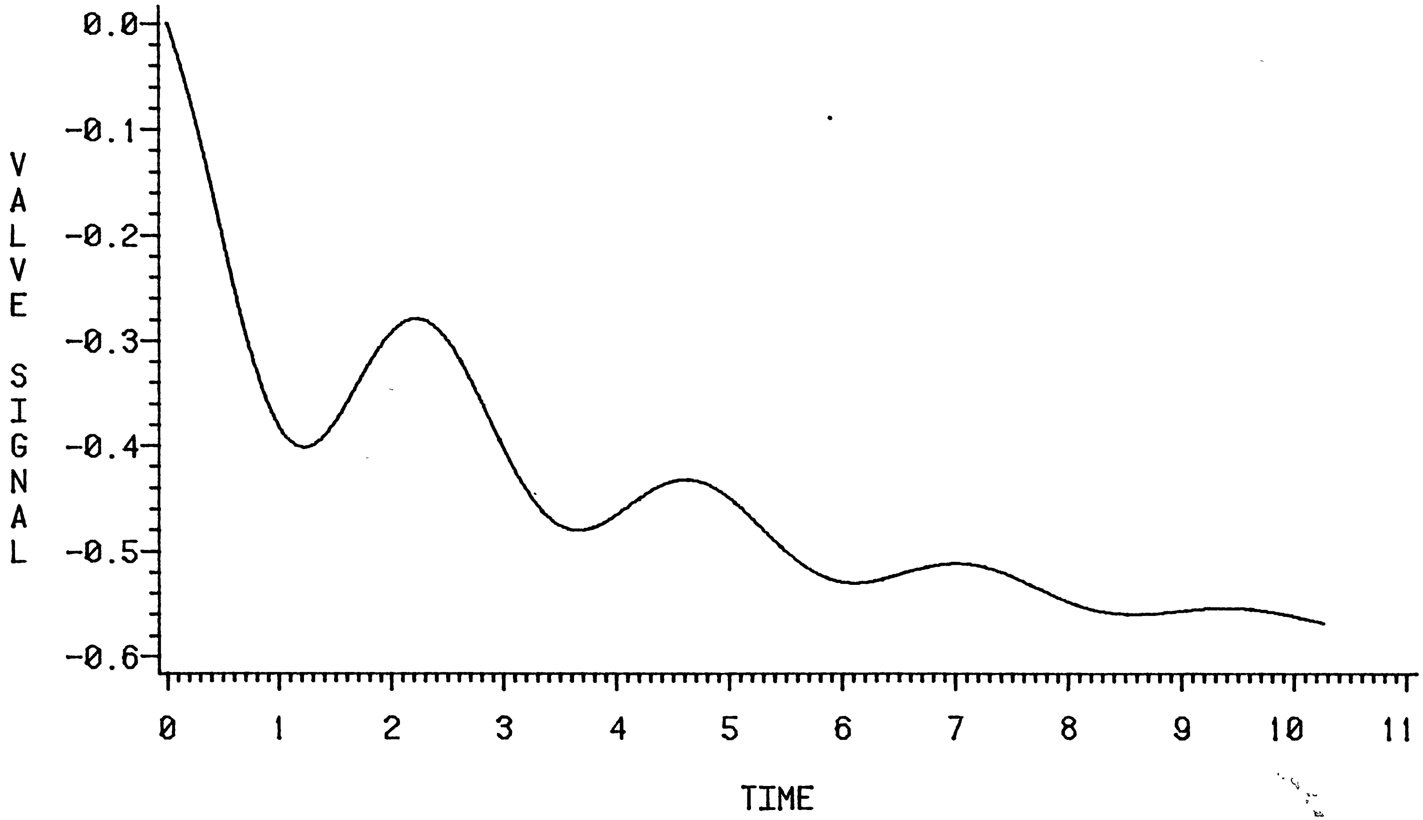


Figure 64. Valve Signal, PI Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 4.0$

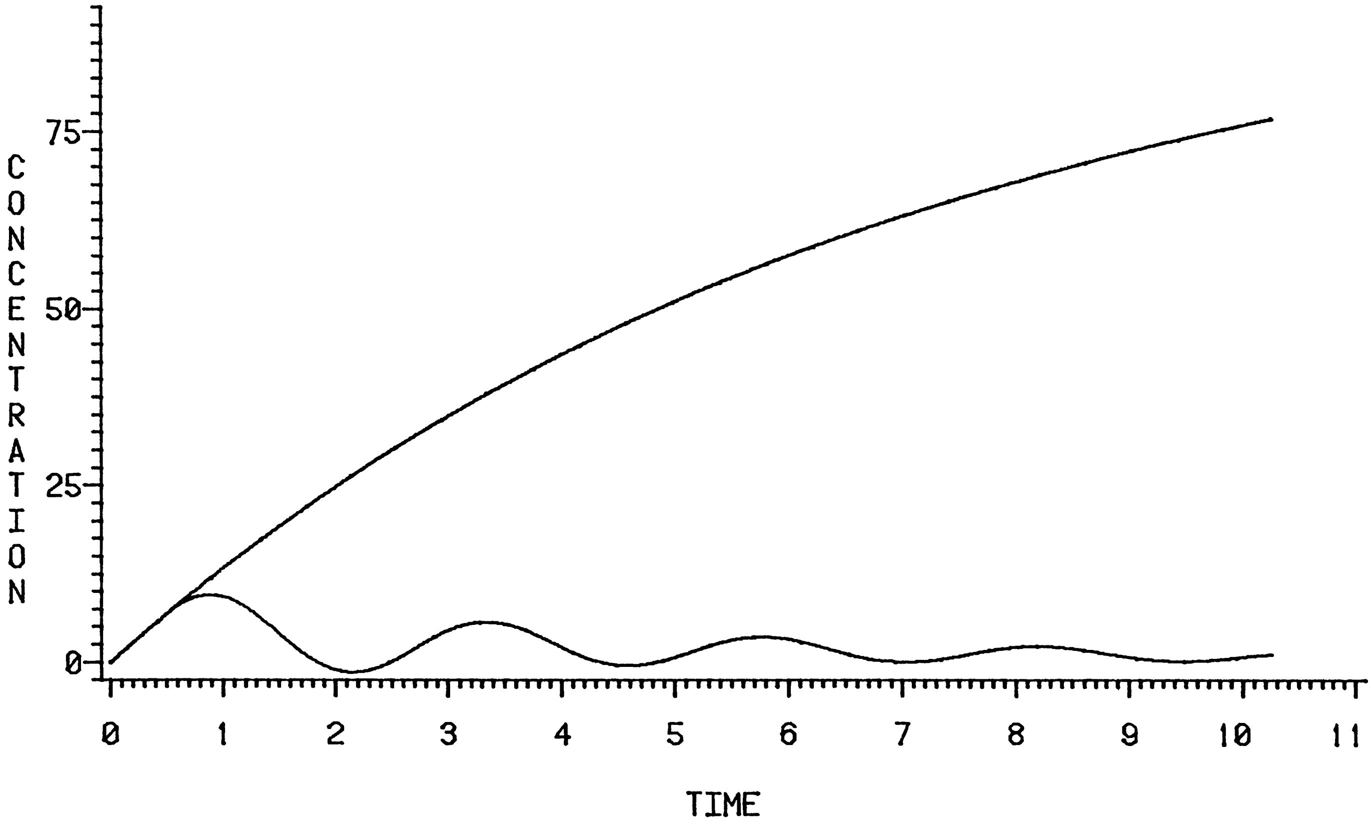


Figure 65. Concentration Variables, θ_d/τ_1 = 0.5, τ_2/τ_1 = 7.0

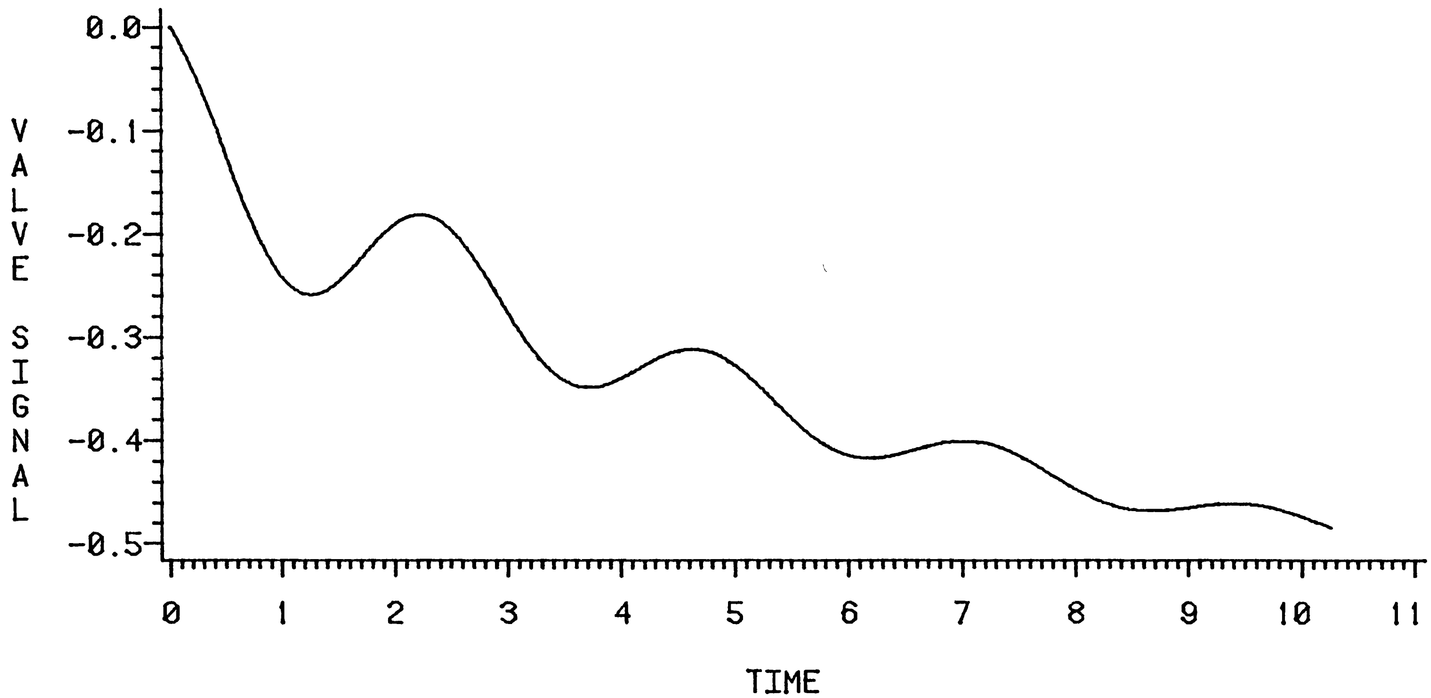


Figure 66. Valve Signal, PI Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 7.0

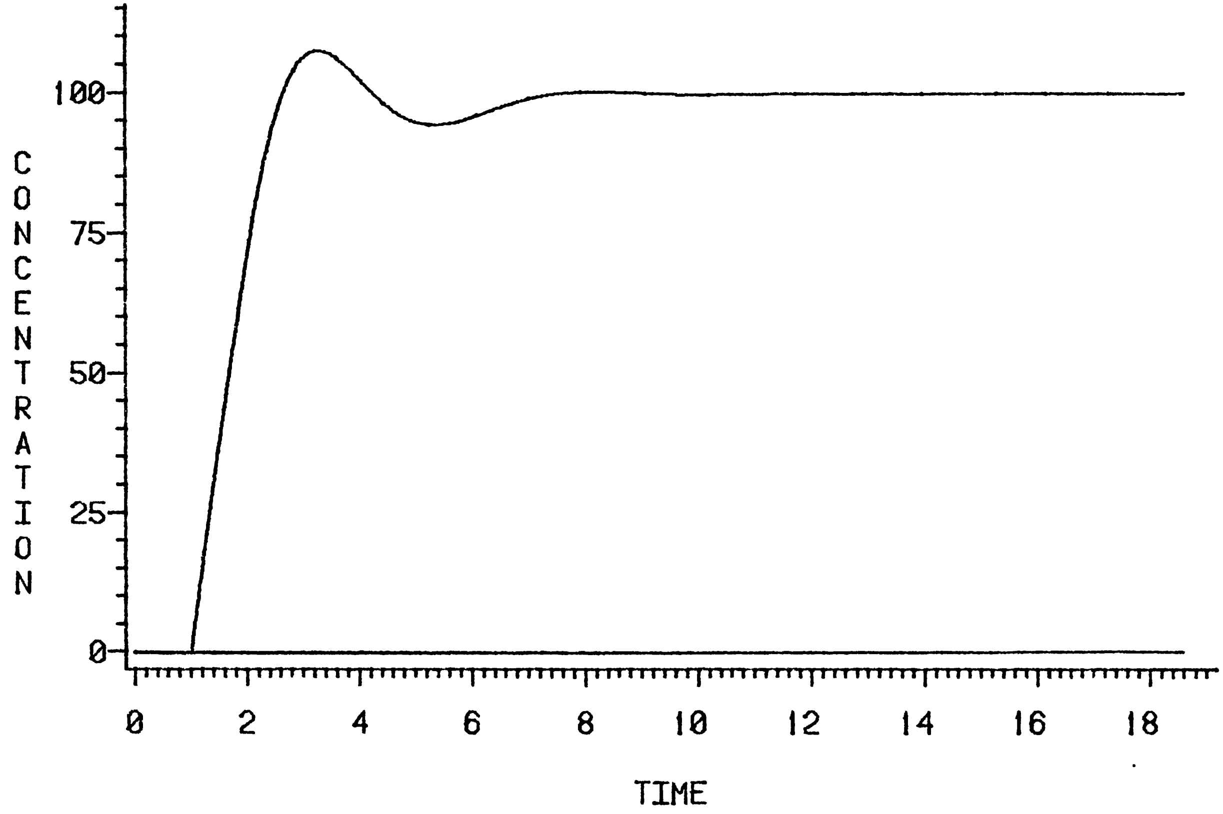


Figure 67. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 1.0$, $r = 0.4$

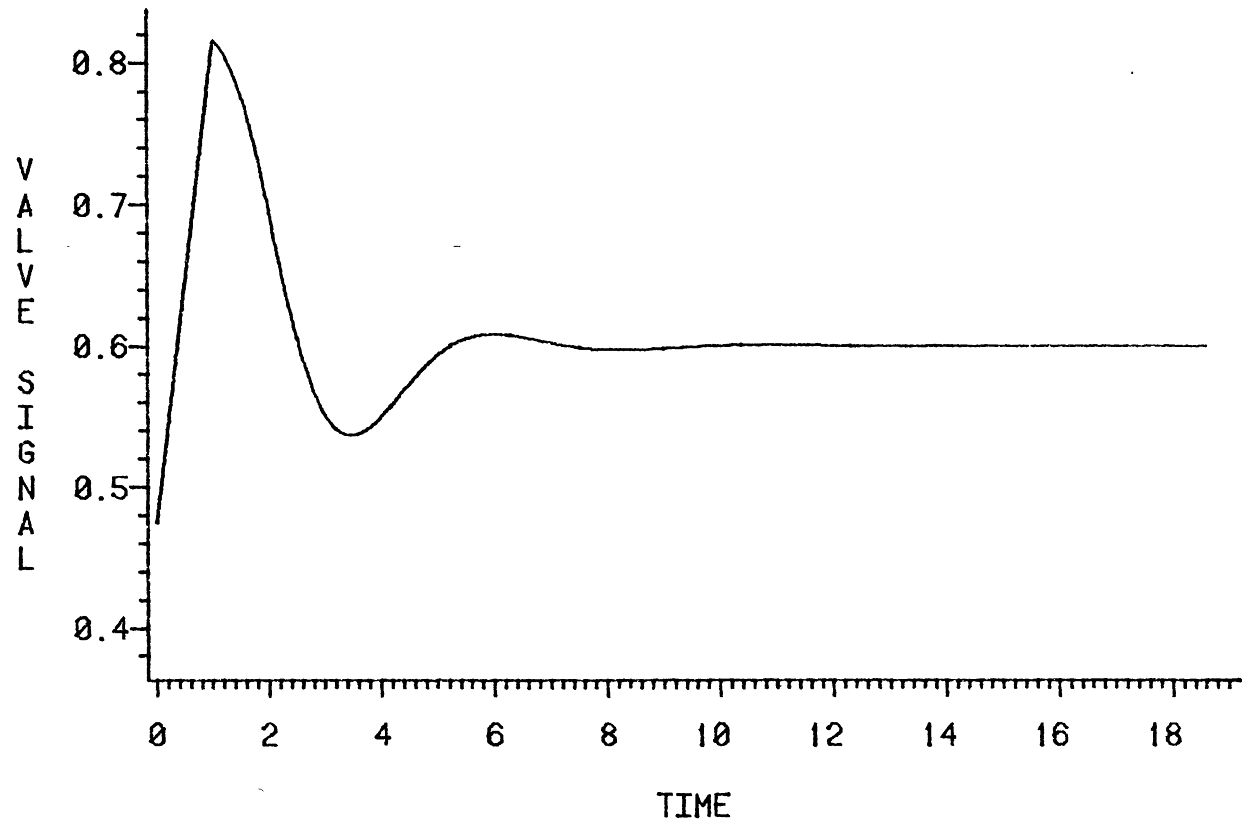


Figure 68. Valve Signal, PI Controller, $\theta_d/\tau_1 = 1.0$, $r = 0.4$

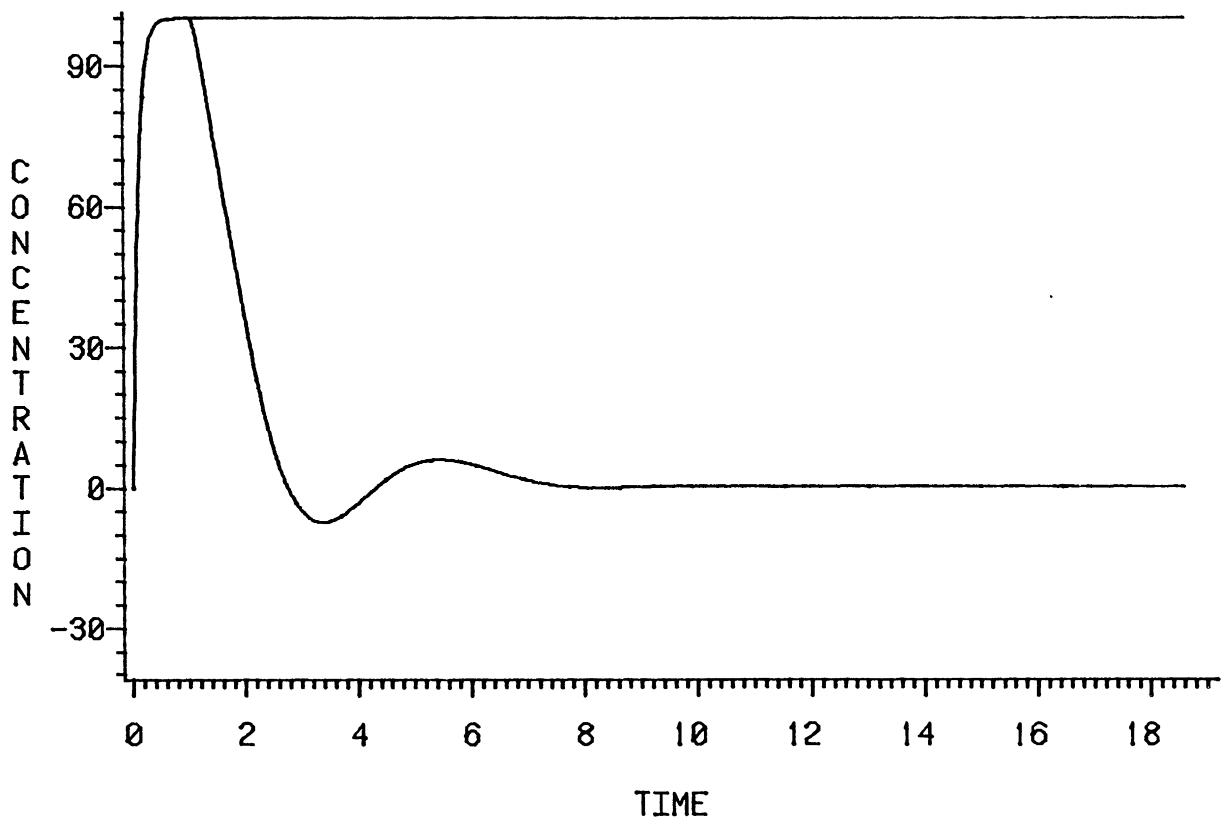


Figure 69. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.1$

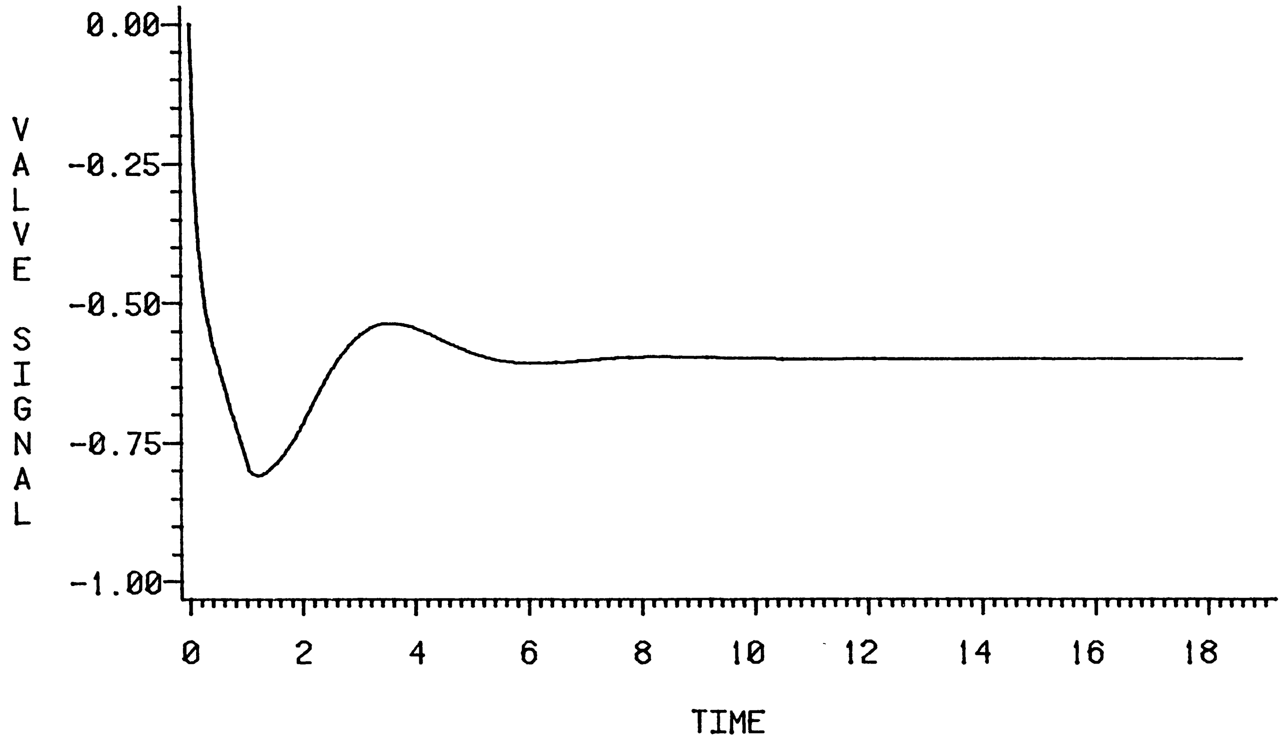


Figure 70. Valve Signal, PI Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.1$

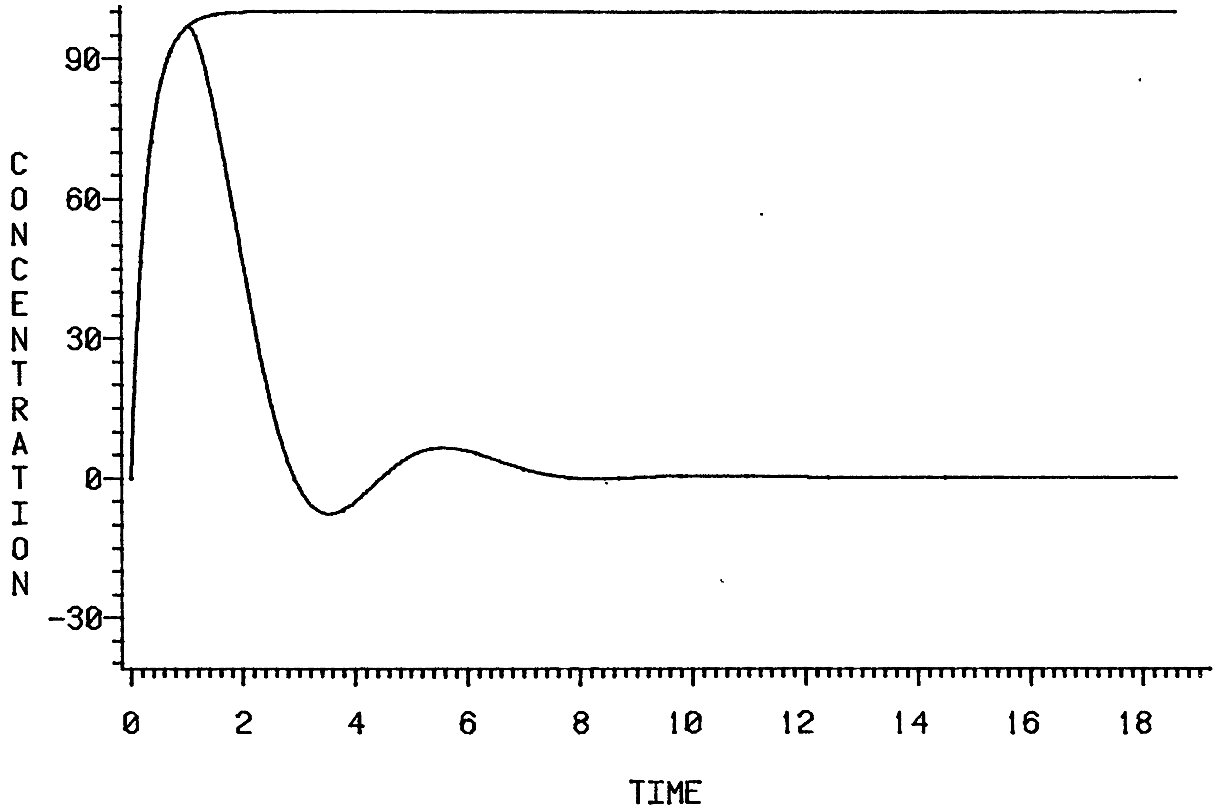


Figure 71. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.3$

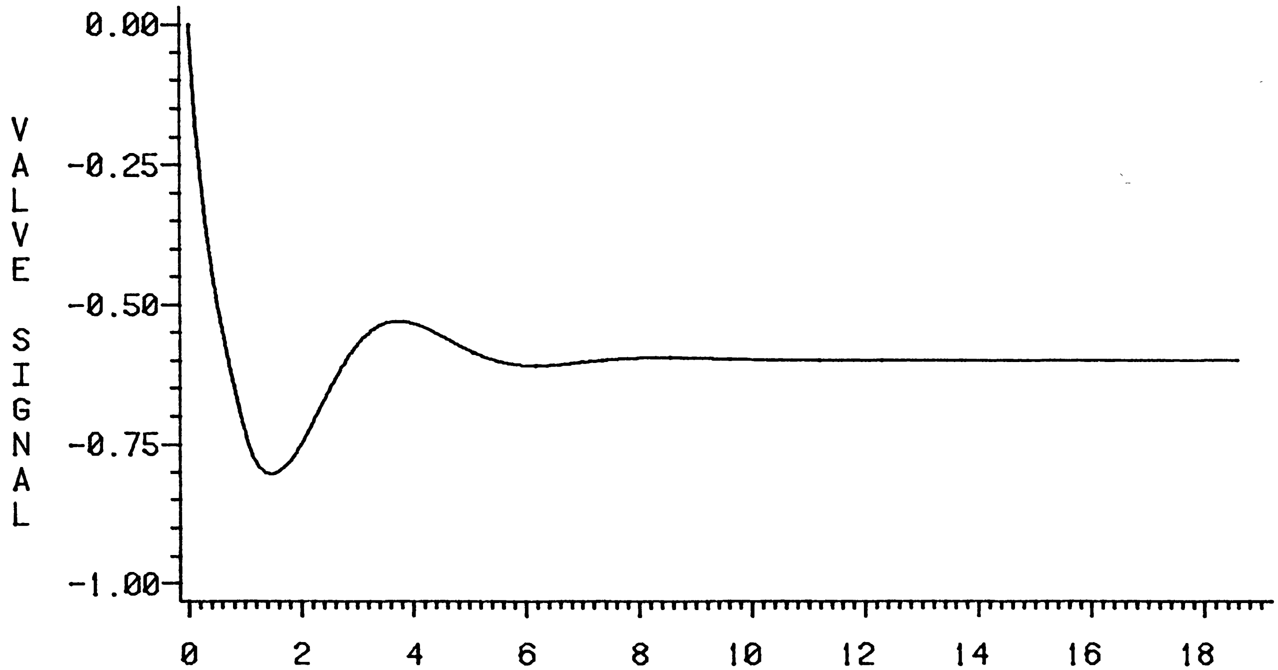


Figure 72. Valve Signal, PI controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 0.3

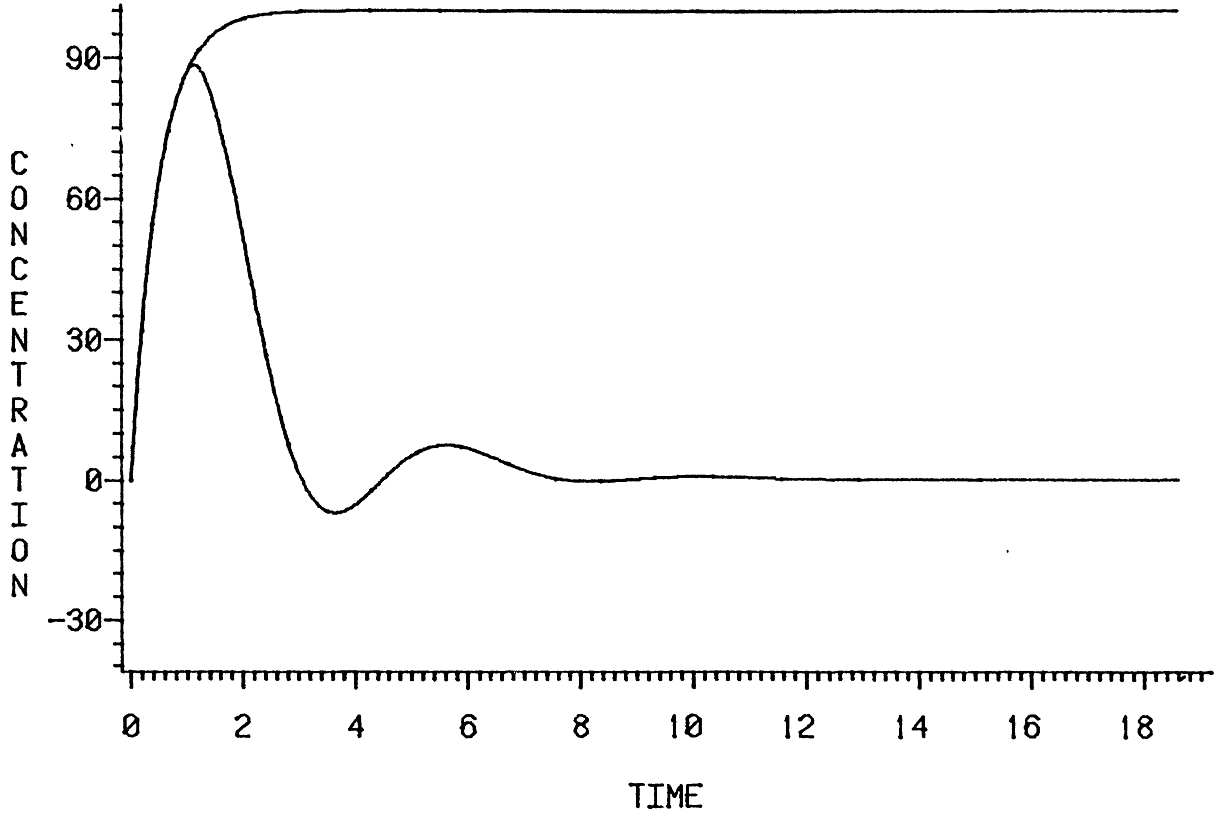


Figure 73. Concentration Variables, PI Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 0.5

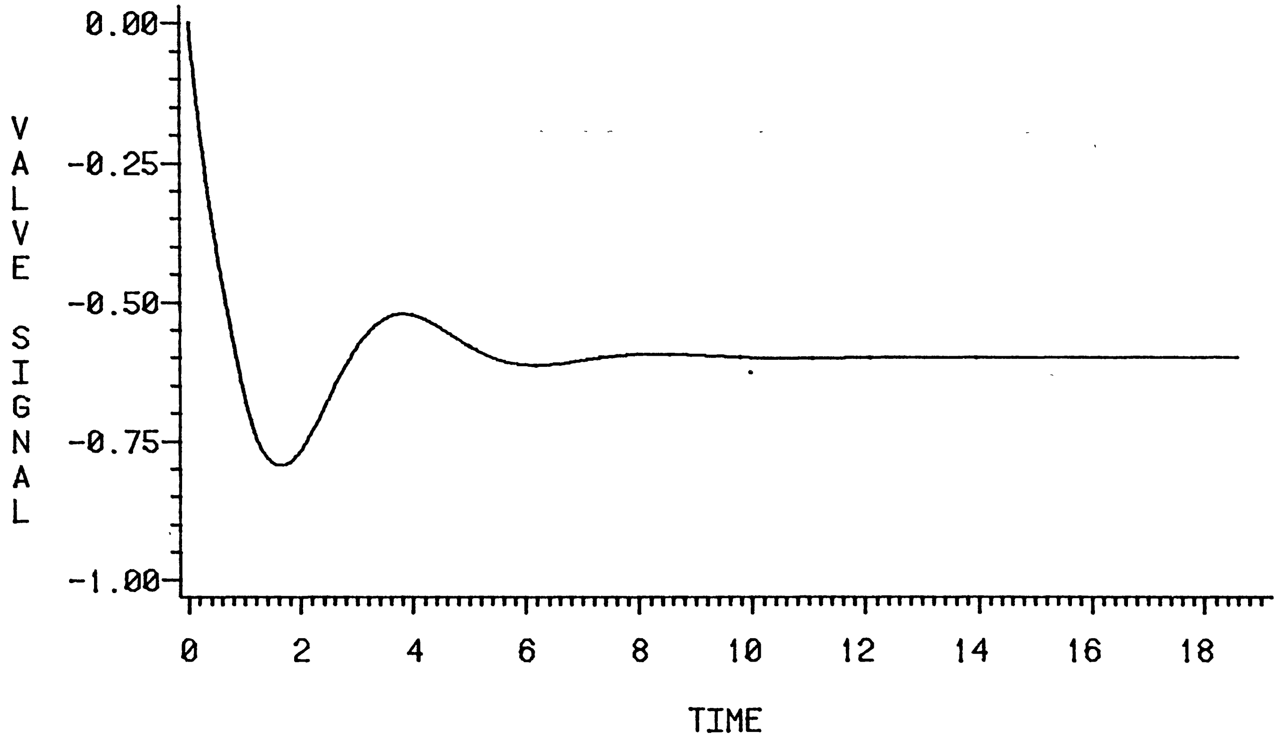


Figure 74. Valve Signal, PI Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.5$

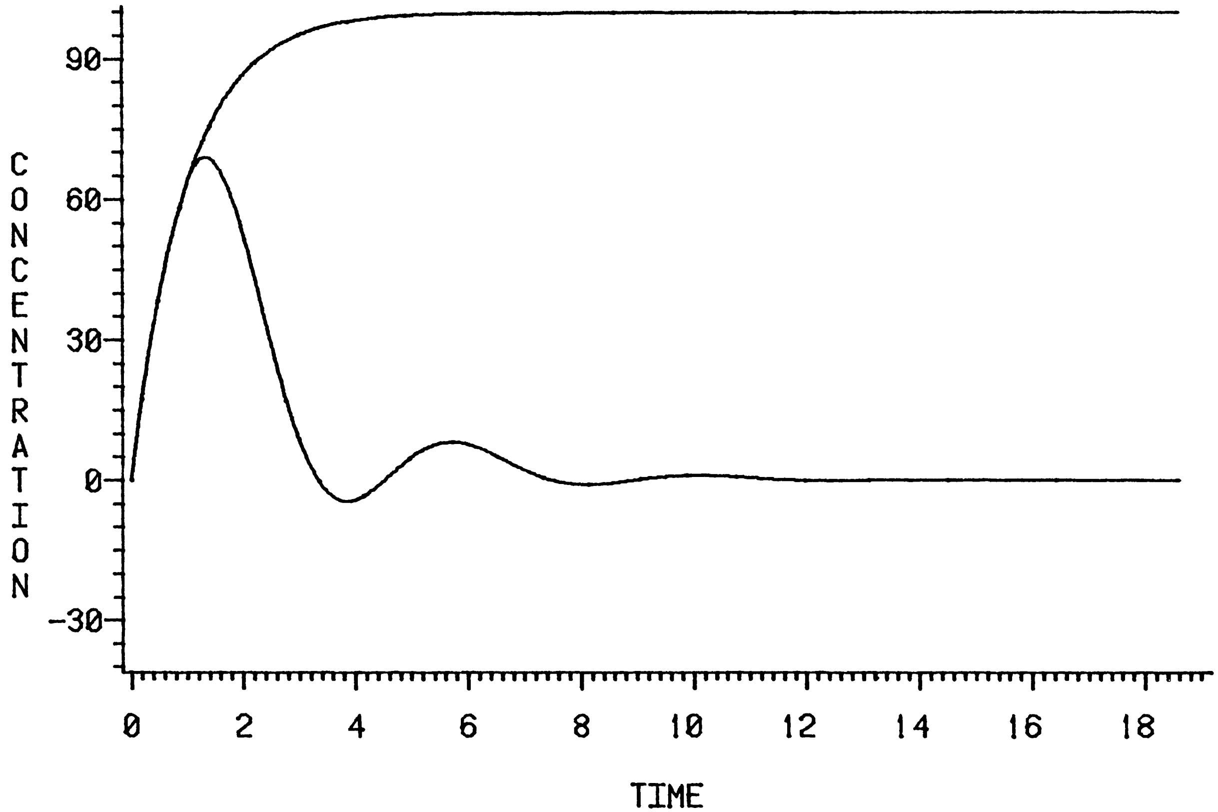


Figure 75. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 1.0$

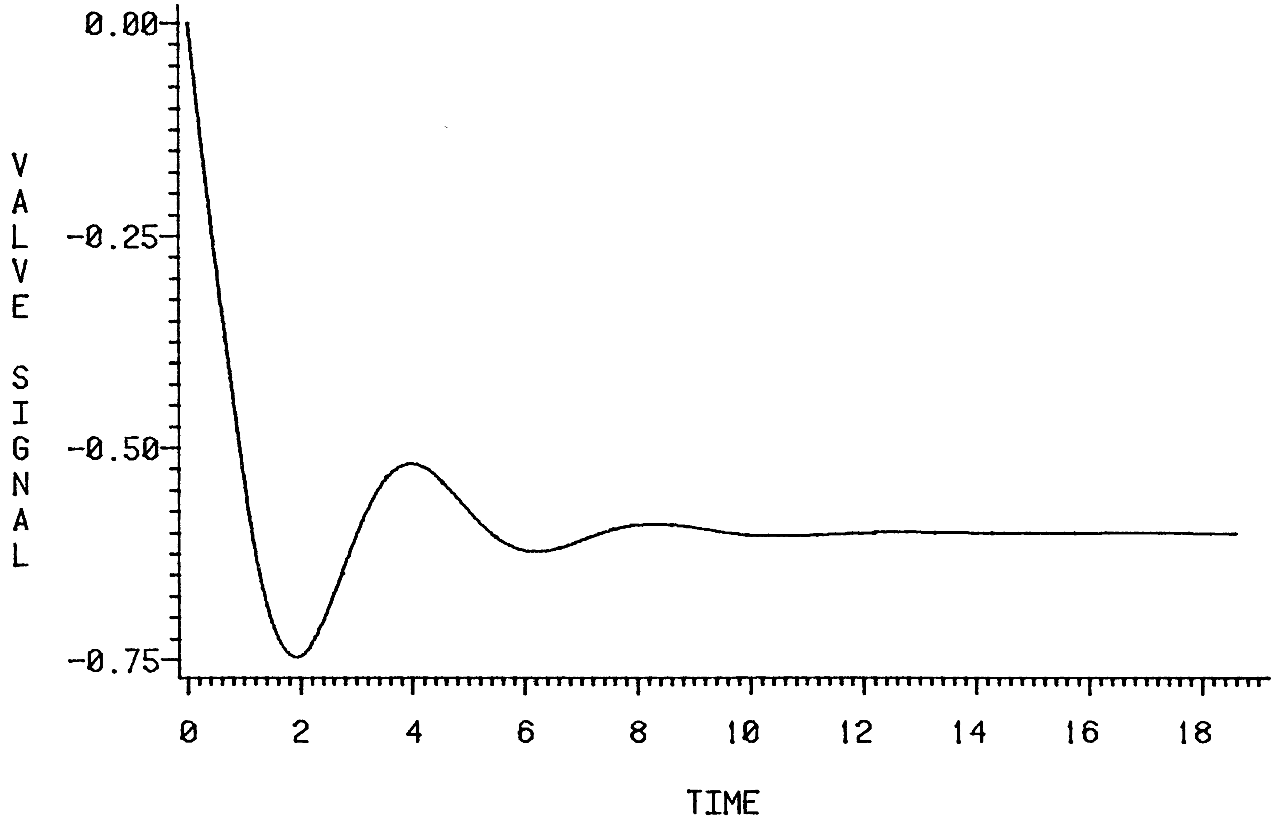


Figure 76. Valve Signal, PI Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 1.0

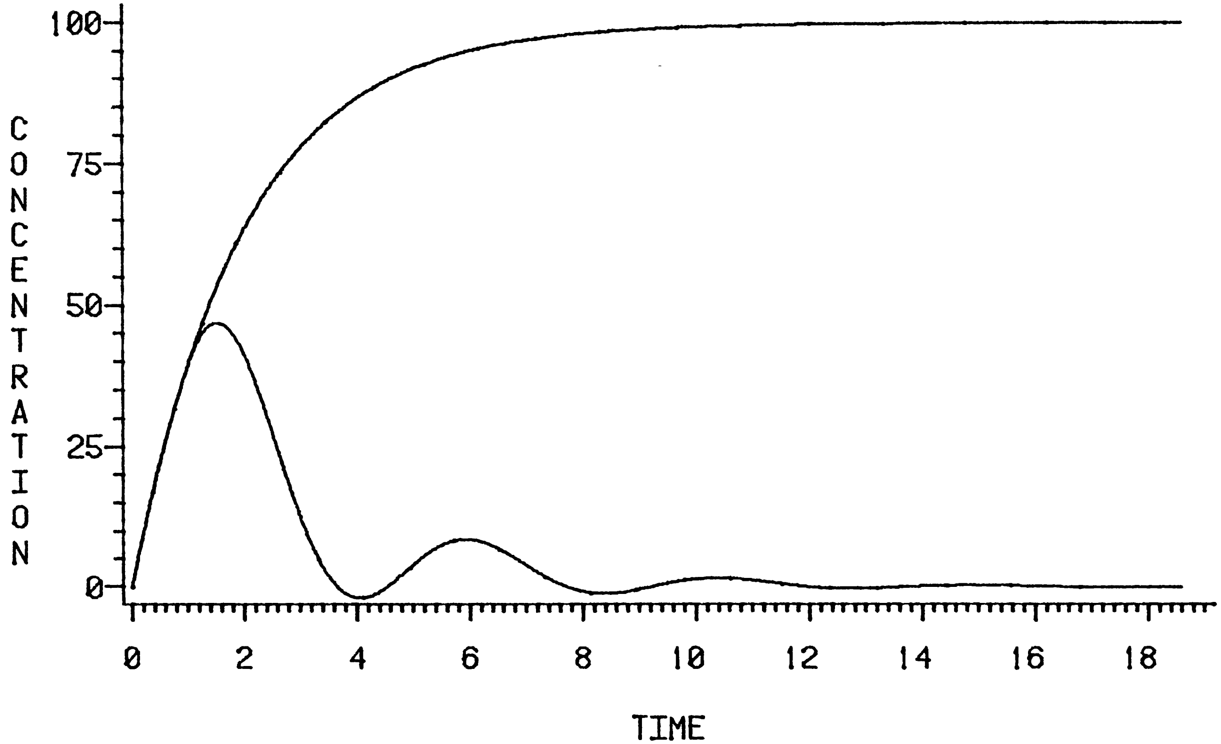


Figure 77. Concentration Variables, PI Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 2.0

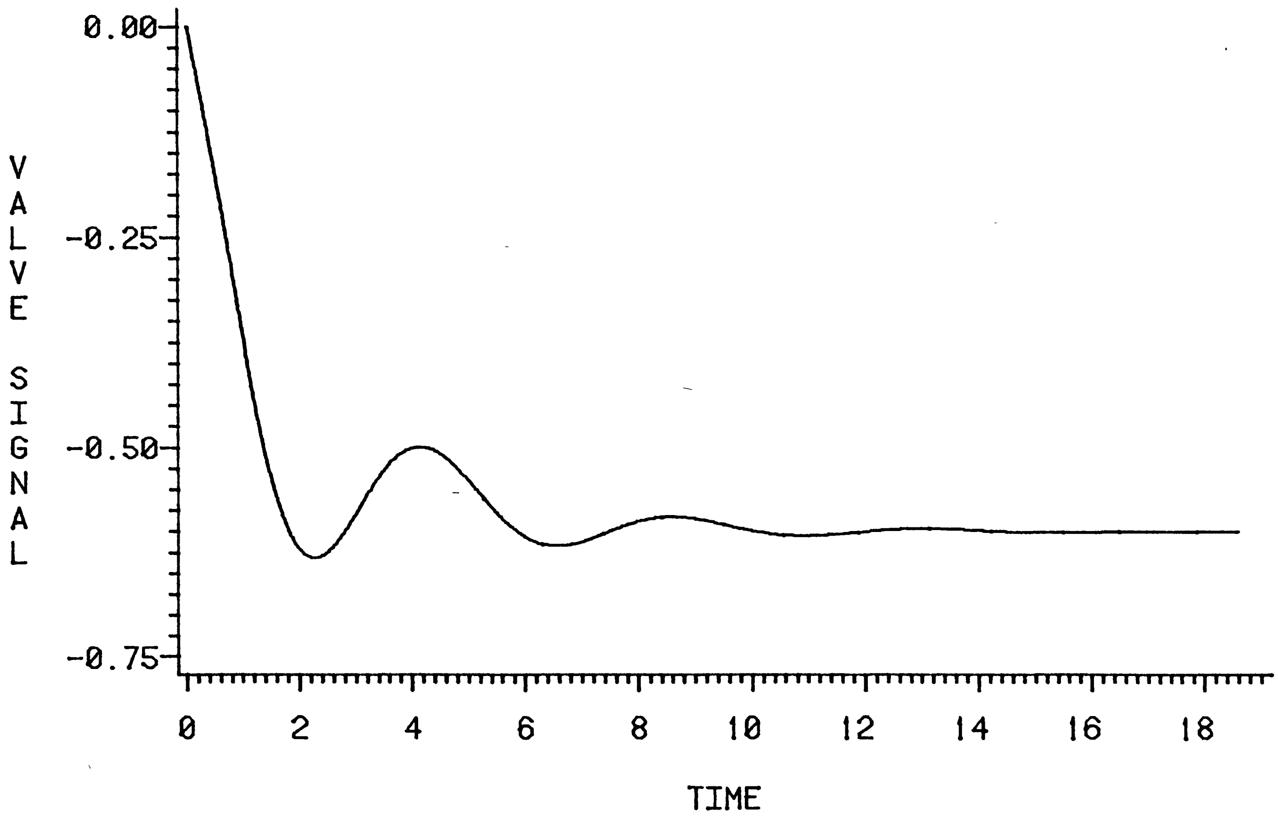


Figure 78. Valve Signal, PI Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 2.0$

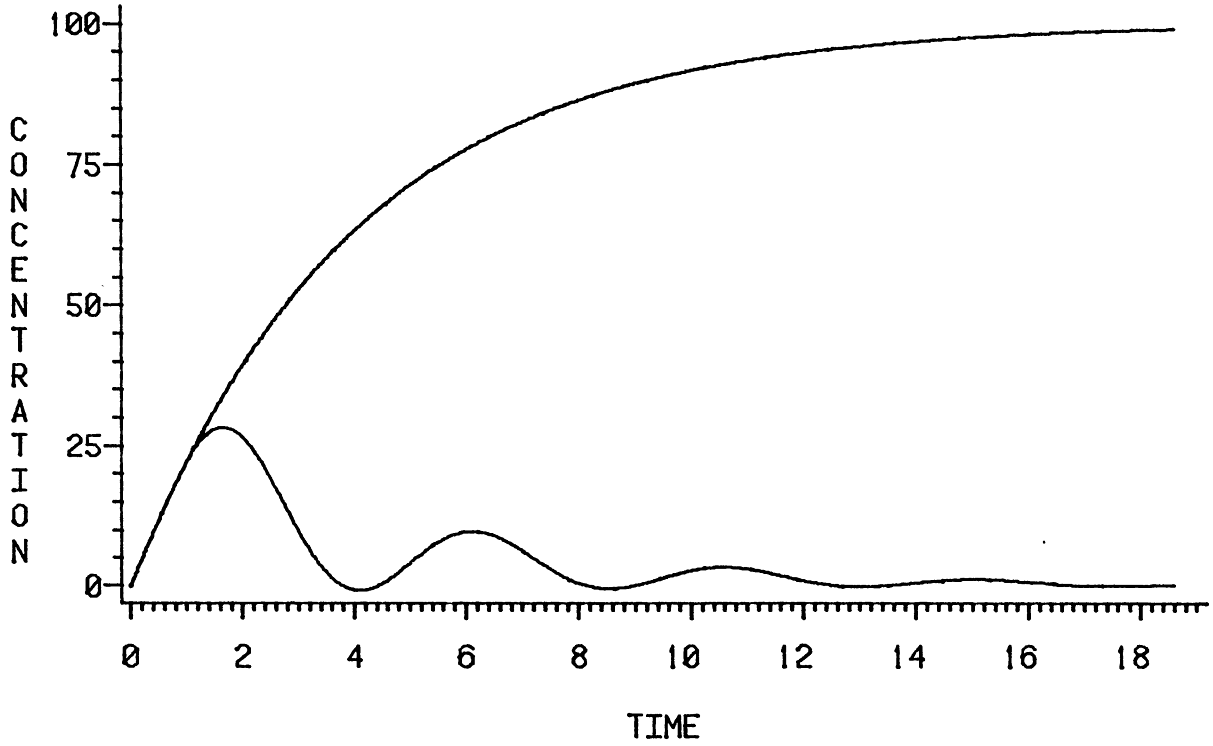


Figure 79. Concentration Variables, PI Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 4.0$

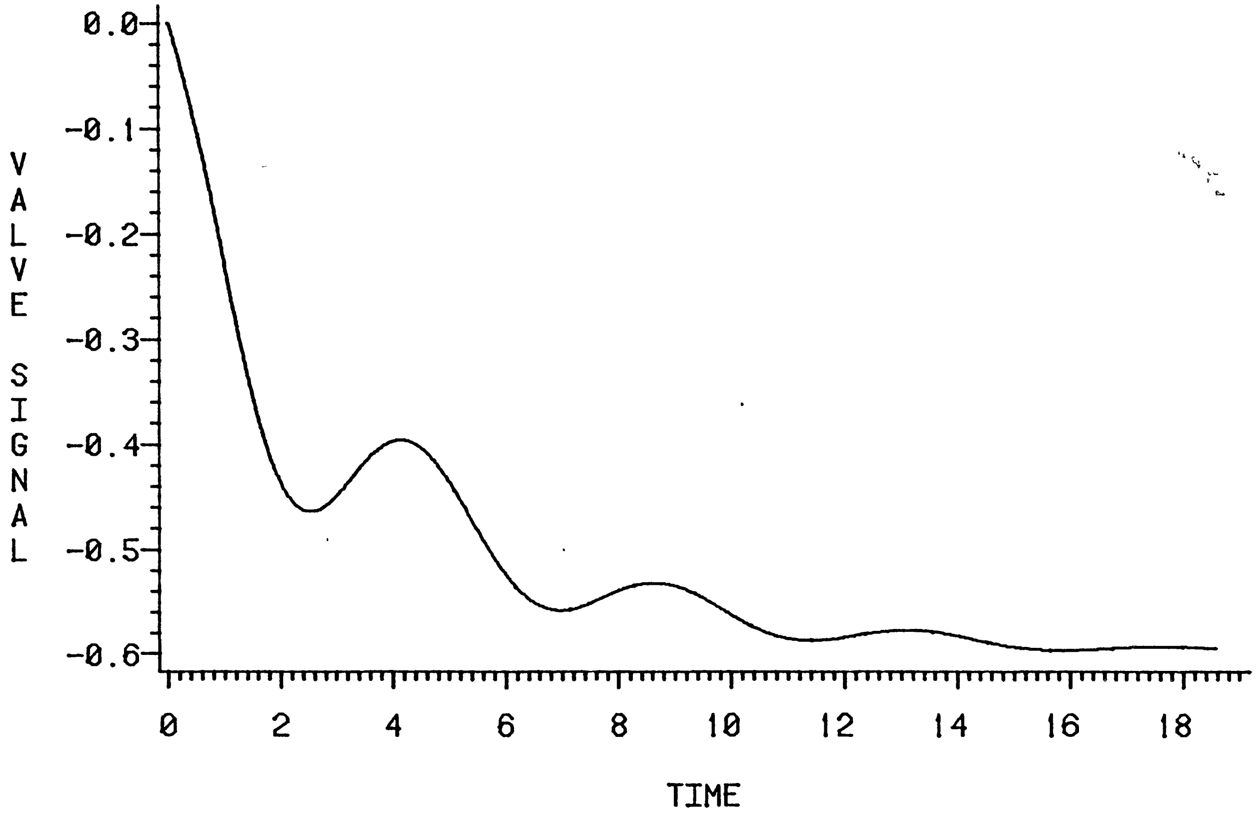


Figure 80. Valve Signal, PI Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 4.0

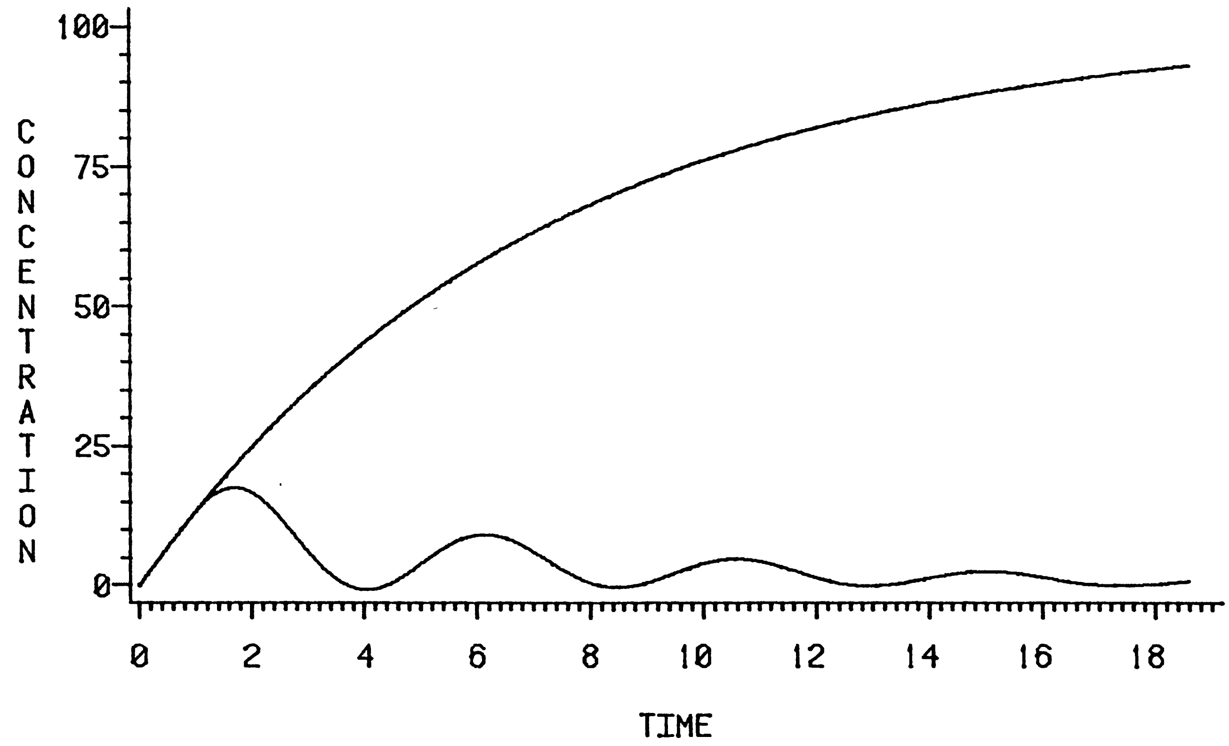


Figure 81. Concentration Variables, PI Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 7.0

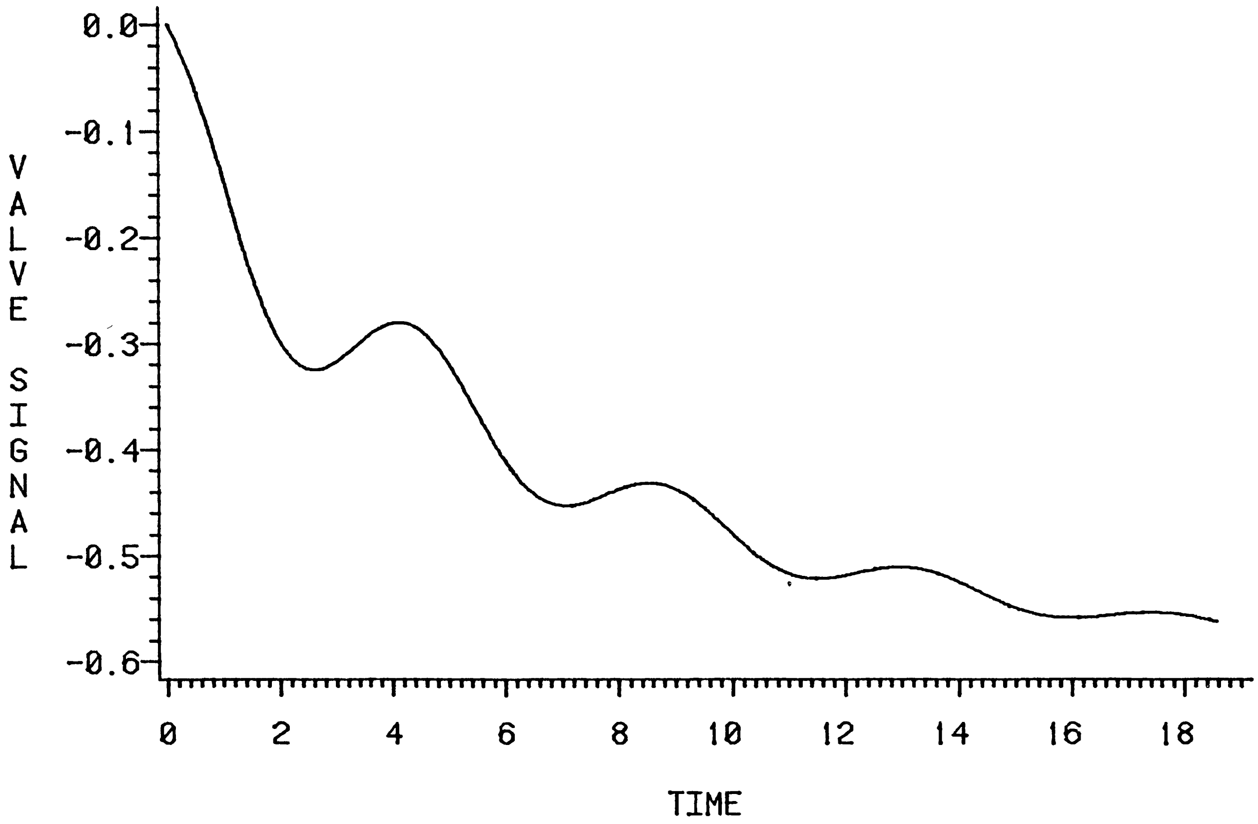


Figure 82. Valve Signal, PI Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 7.0

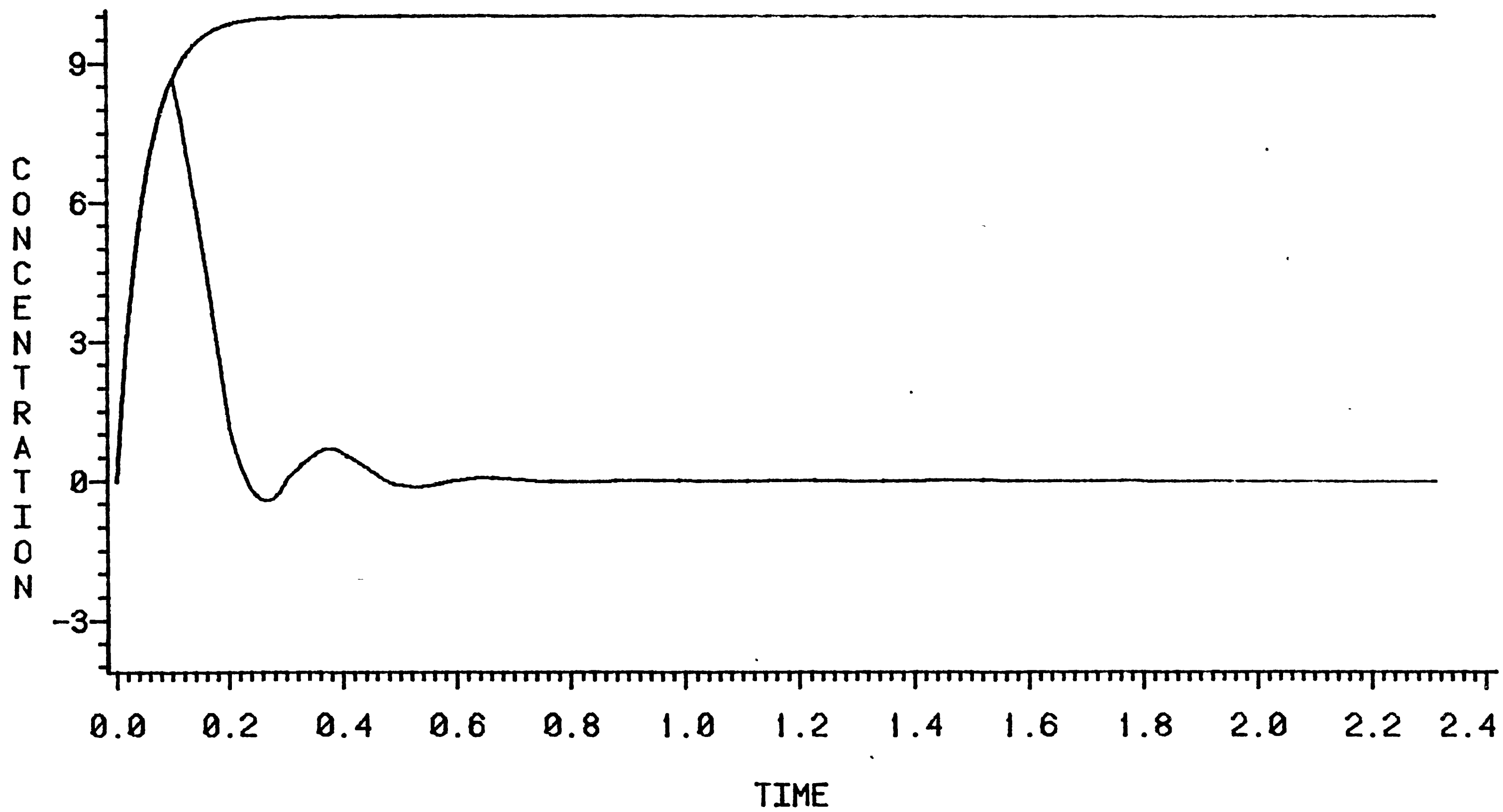


Figure 83. Concentration Variables, PID Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 0.05

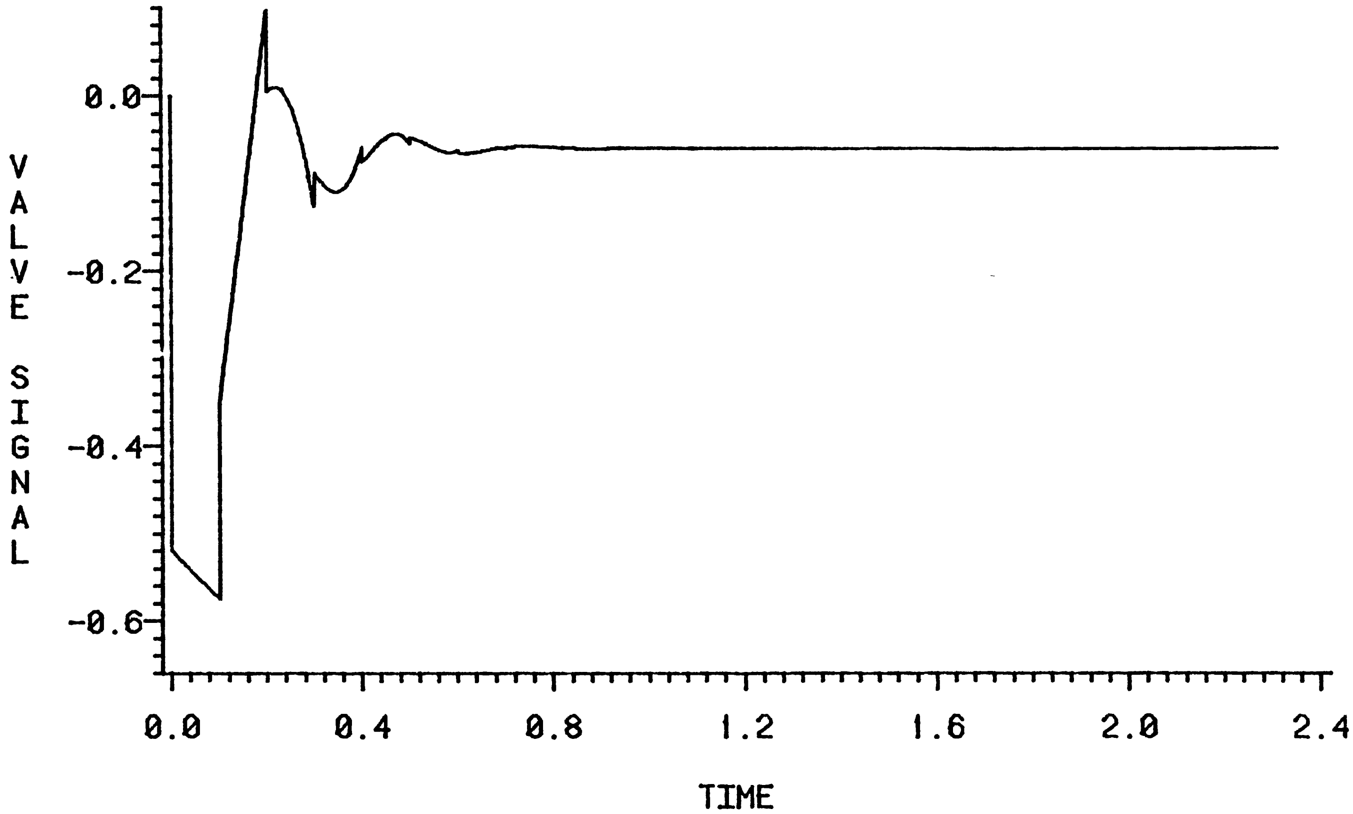


Figure 84. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.05$

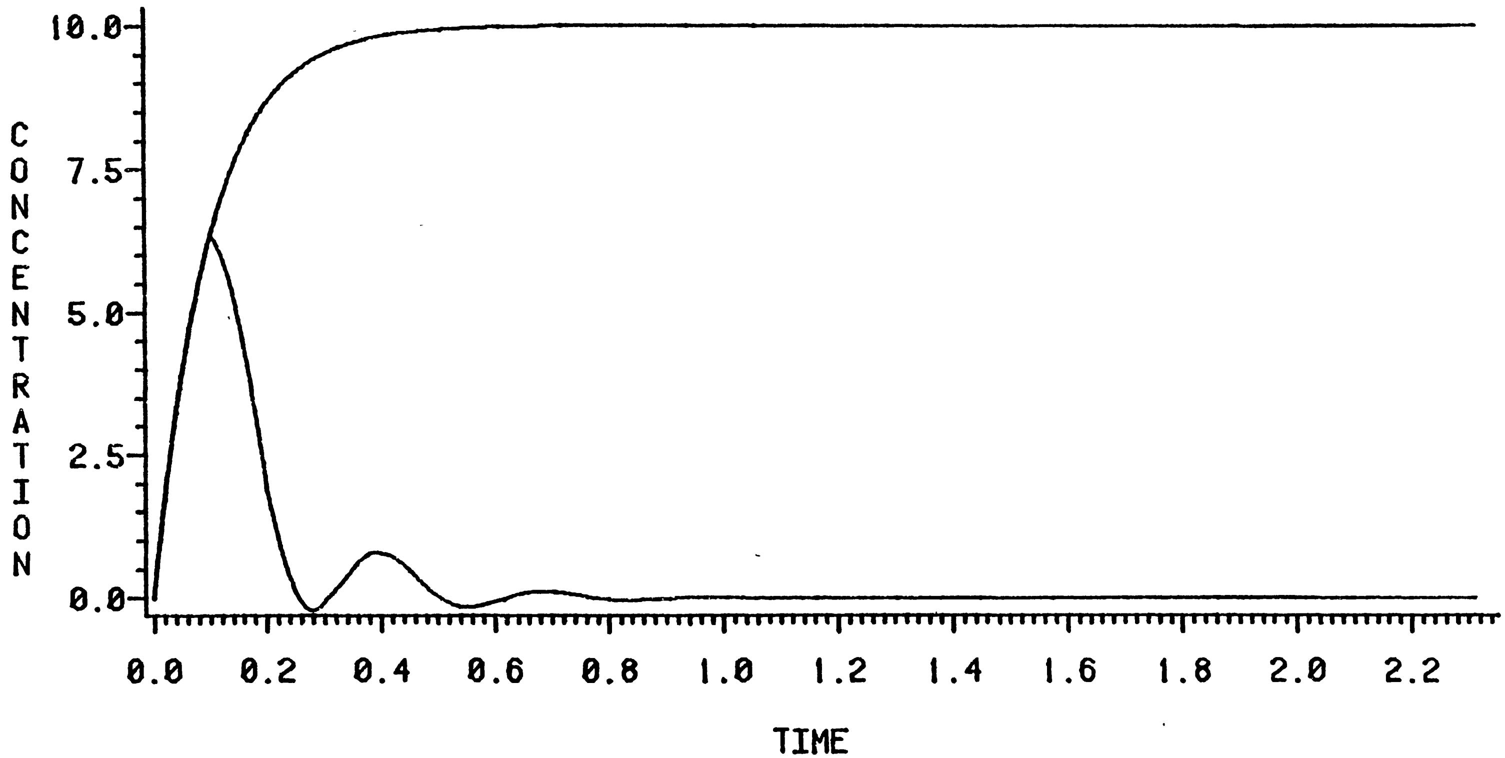


Figure 85. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.1$

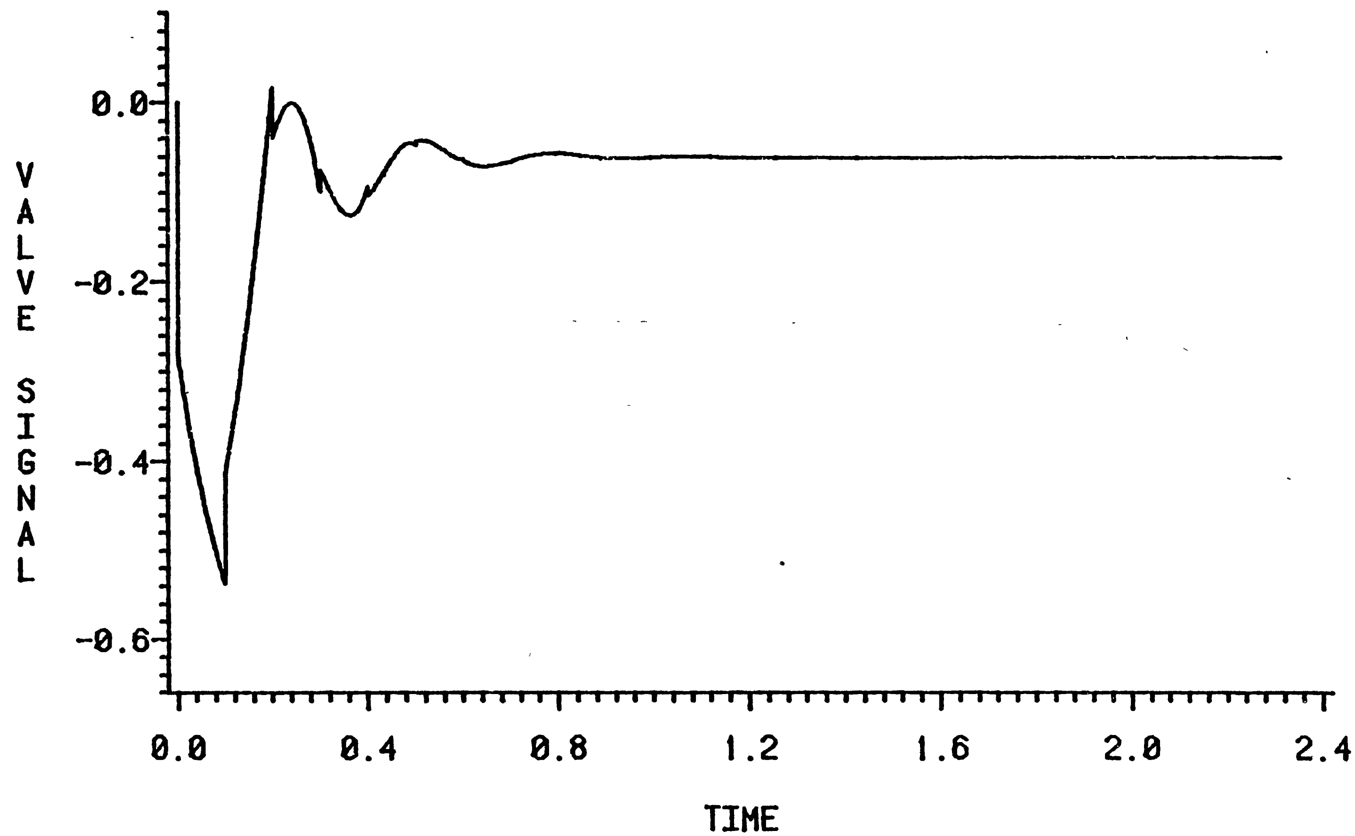


Figure 86. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.1$

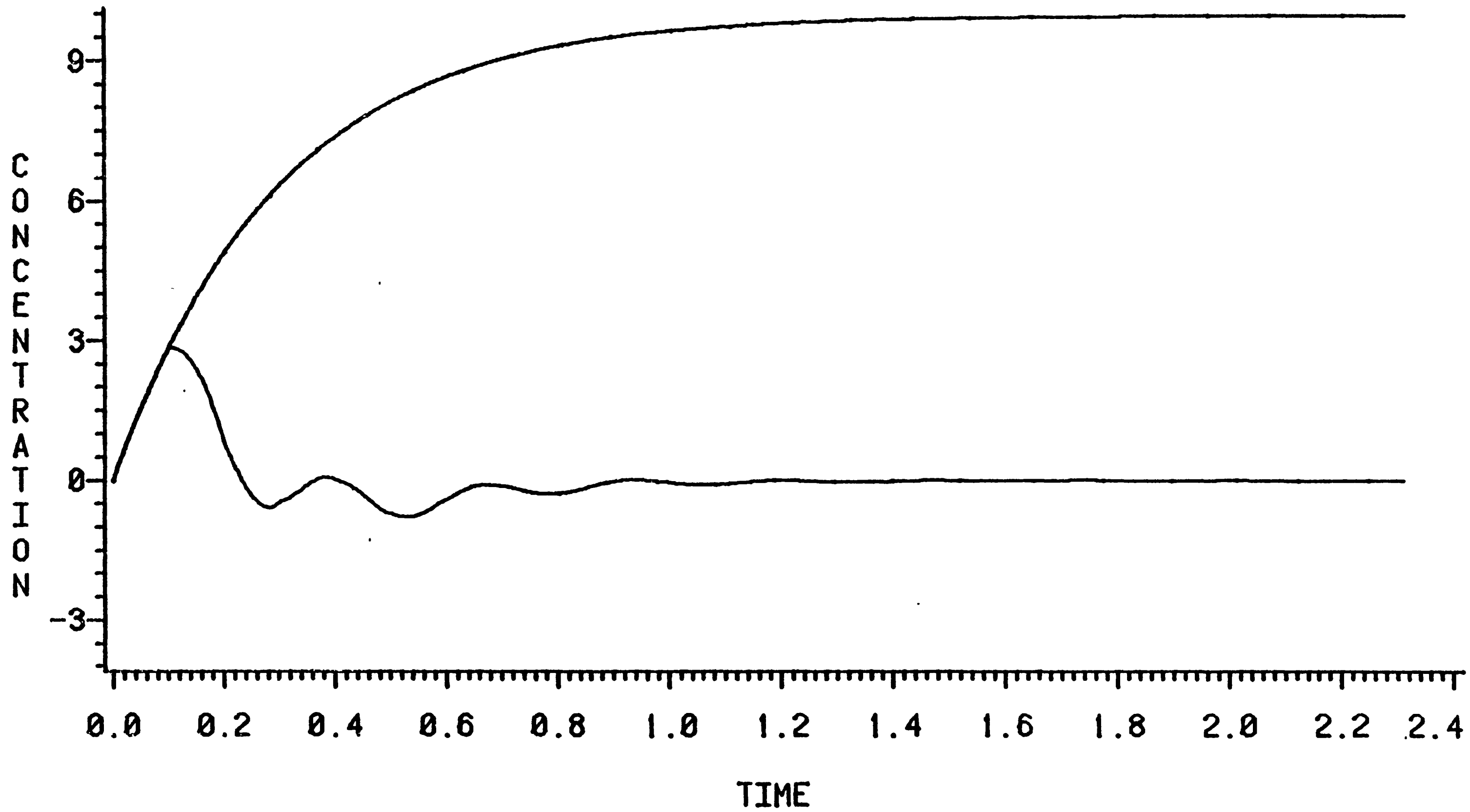


Figure 87. Concentration Variables, PID Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 0.3

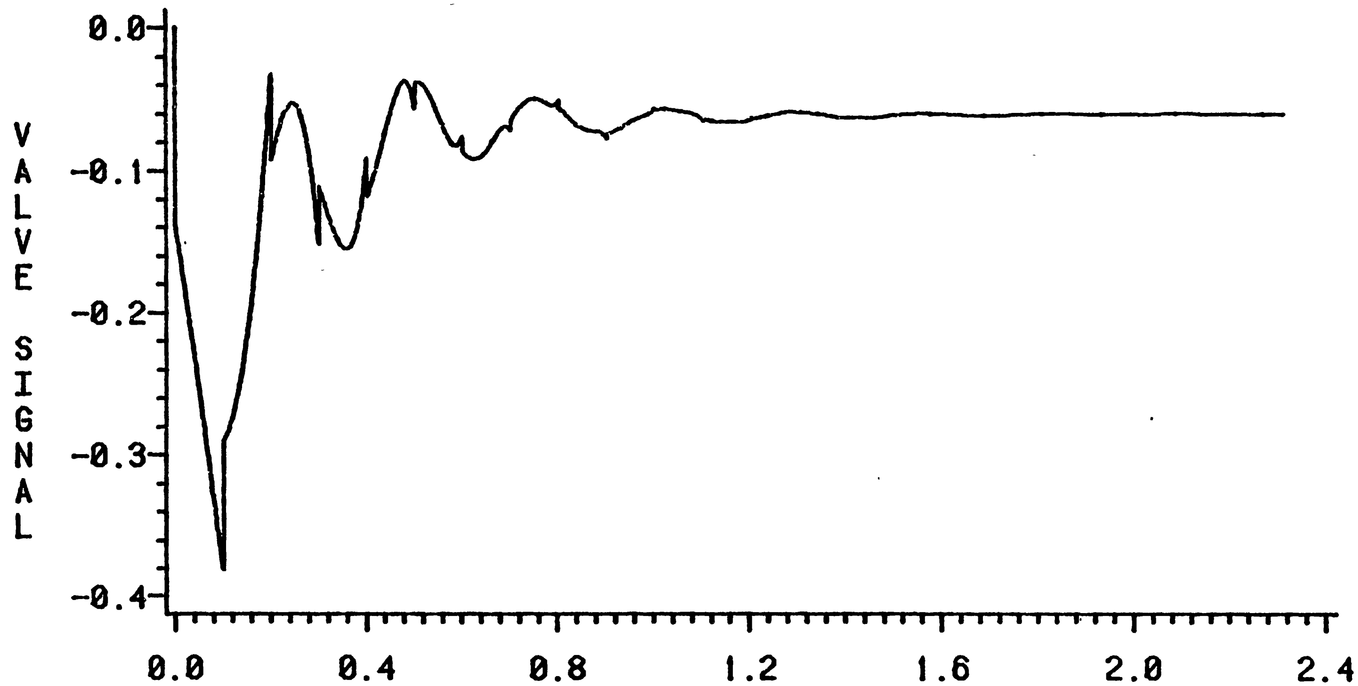


Figure 88. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.3$

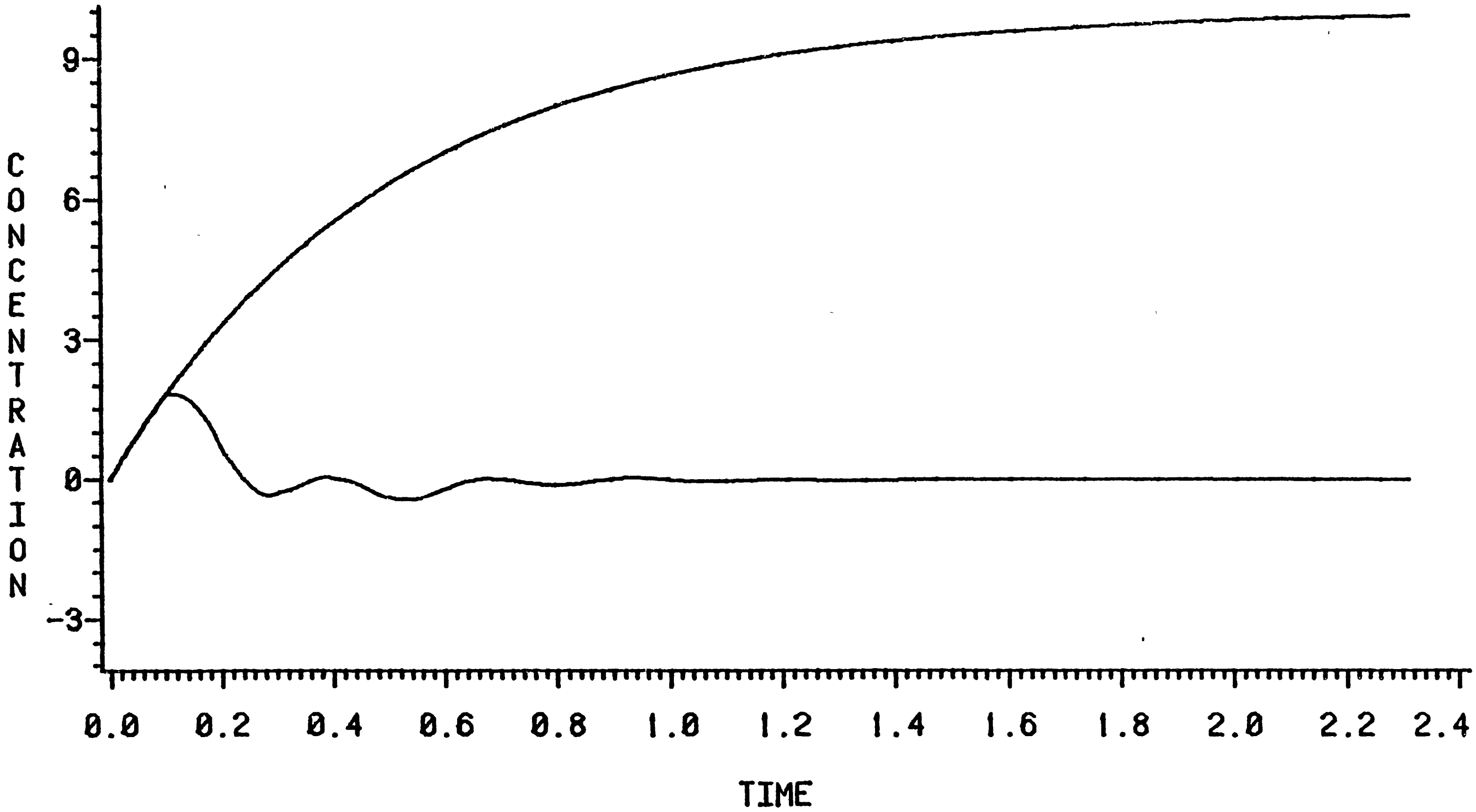


Figure 89. Concentration Variables, PID Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 0.5

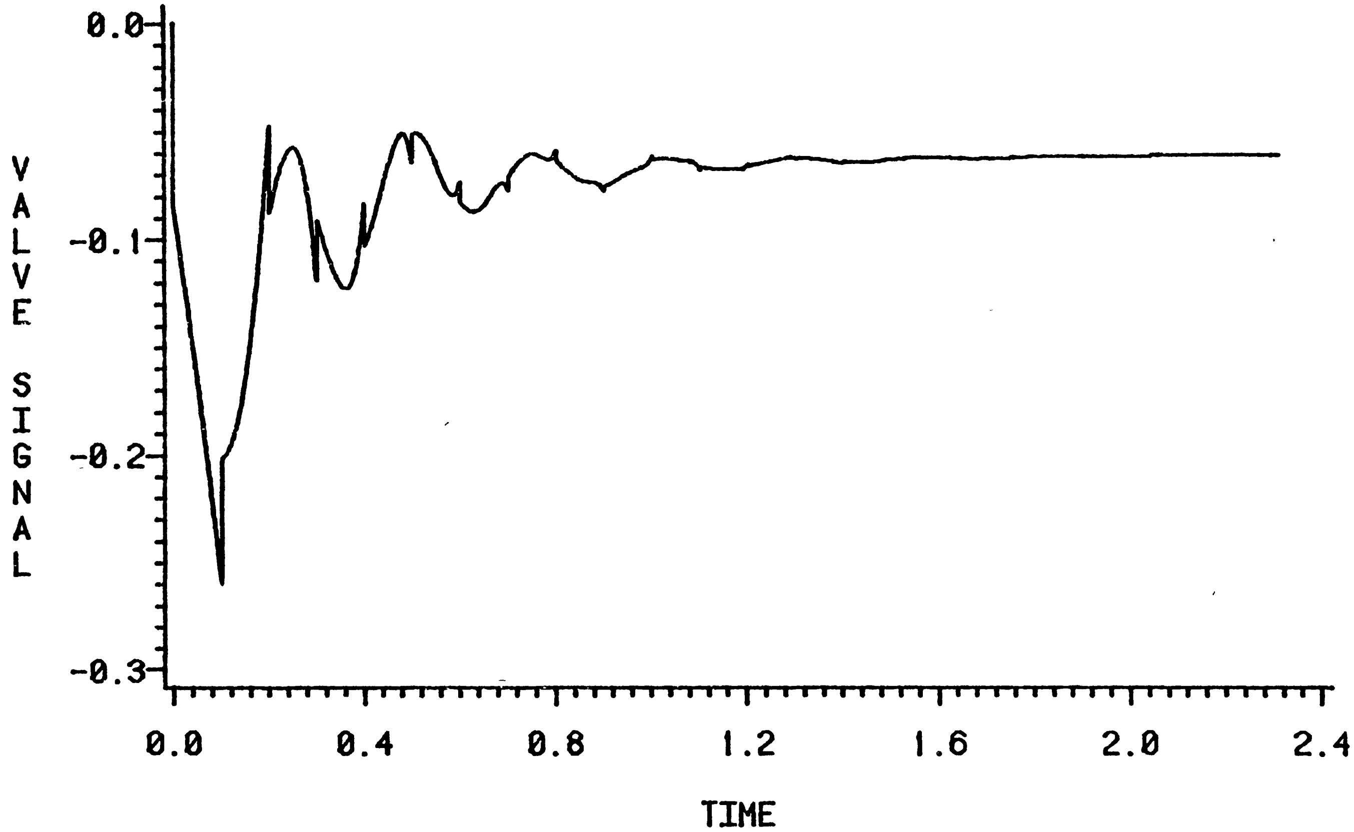


Figure 90. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 0.5$

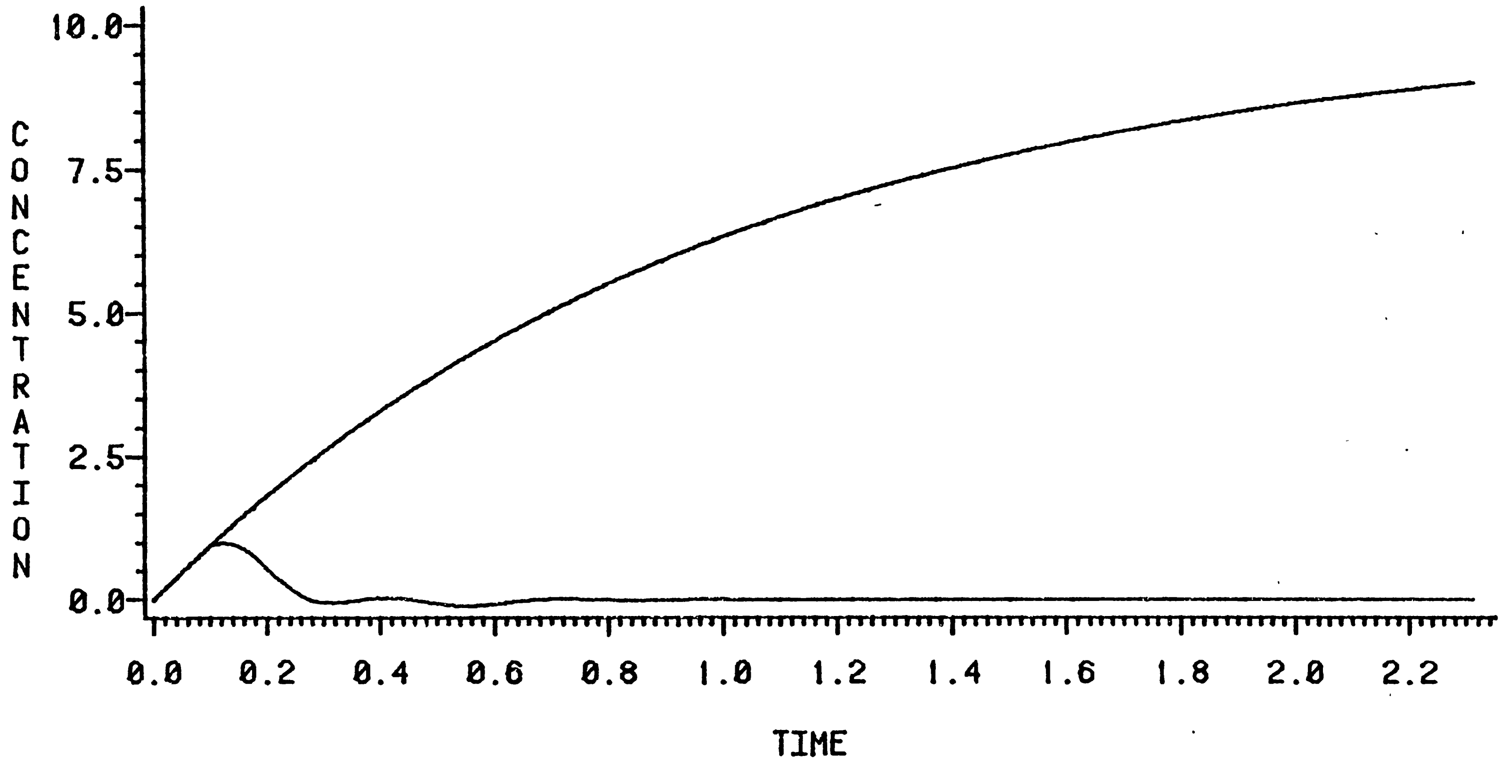


Figure 91. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 1.0$

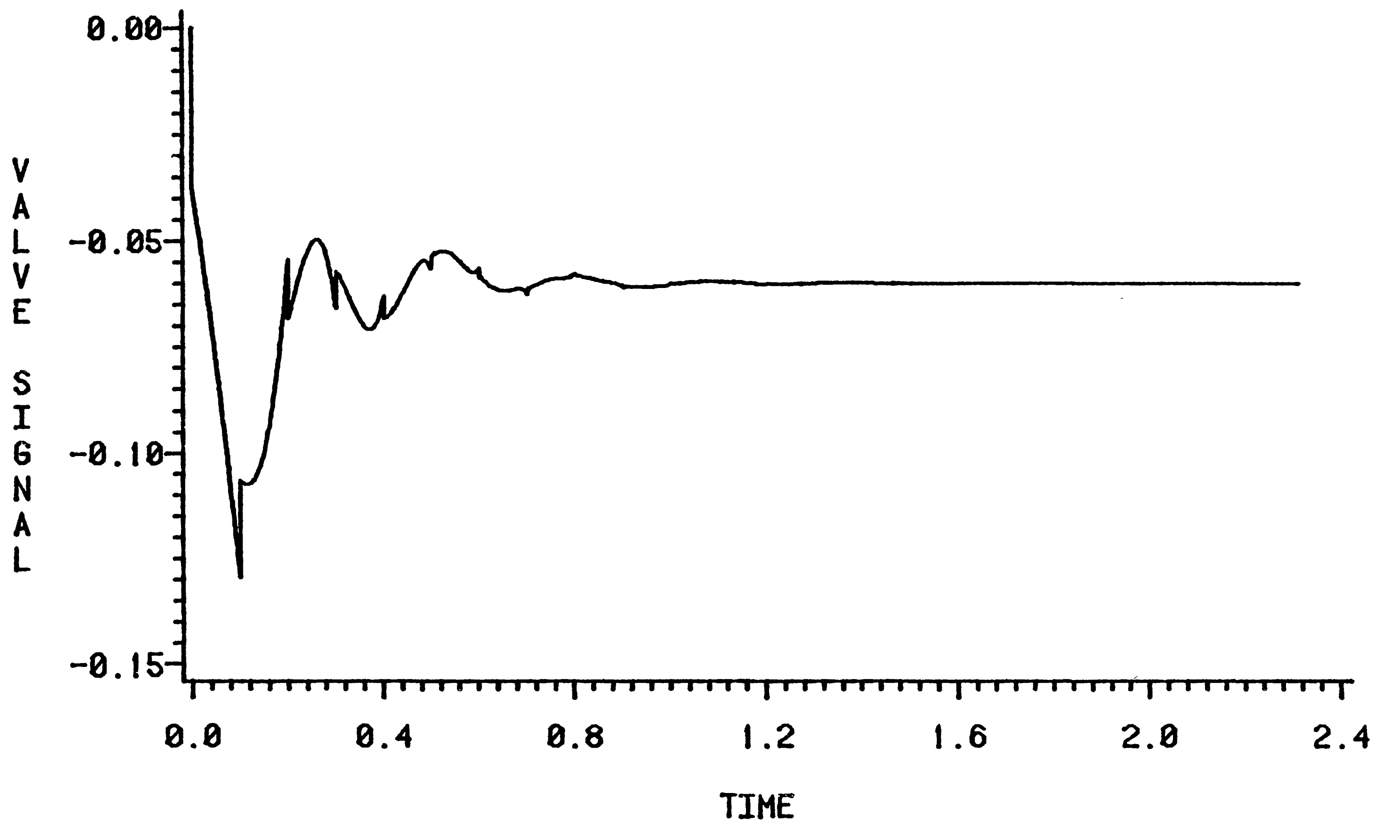


Figure 92. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 1.0$

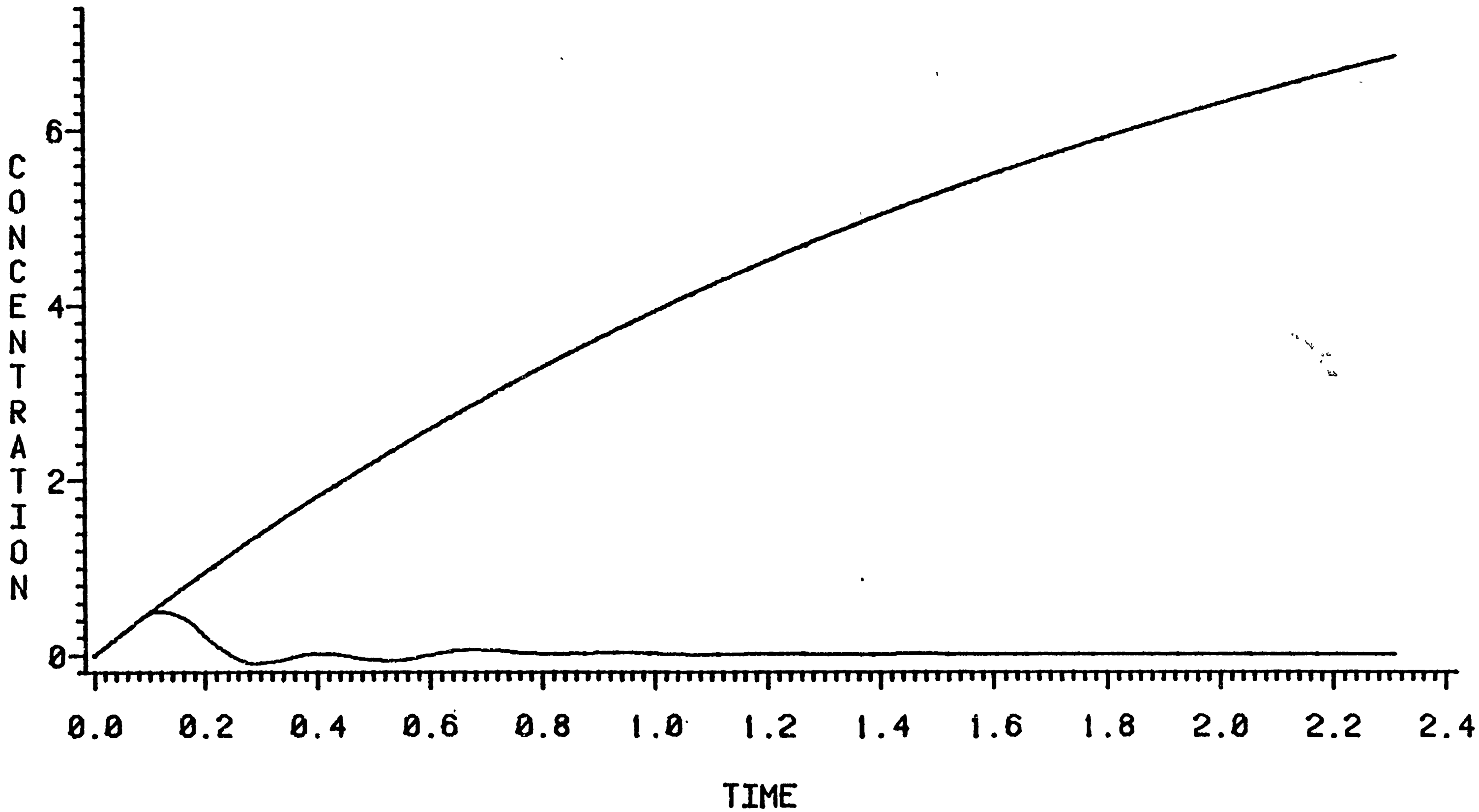


Figure 93. Concentration Variables, PID Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 2.0

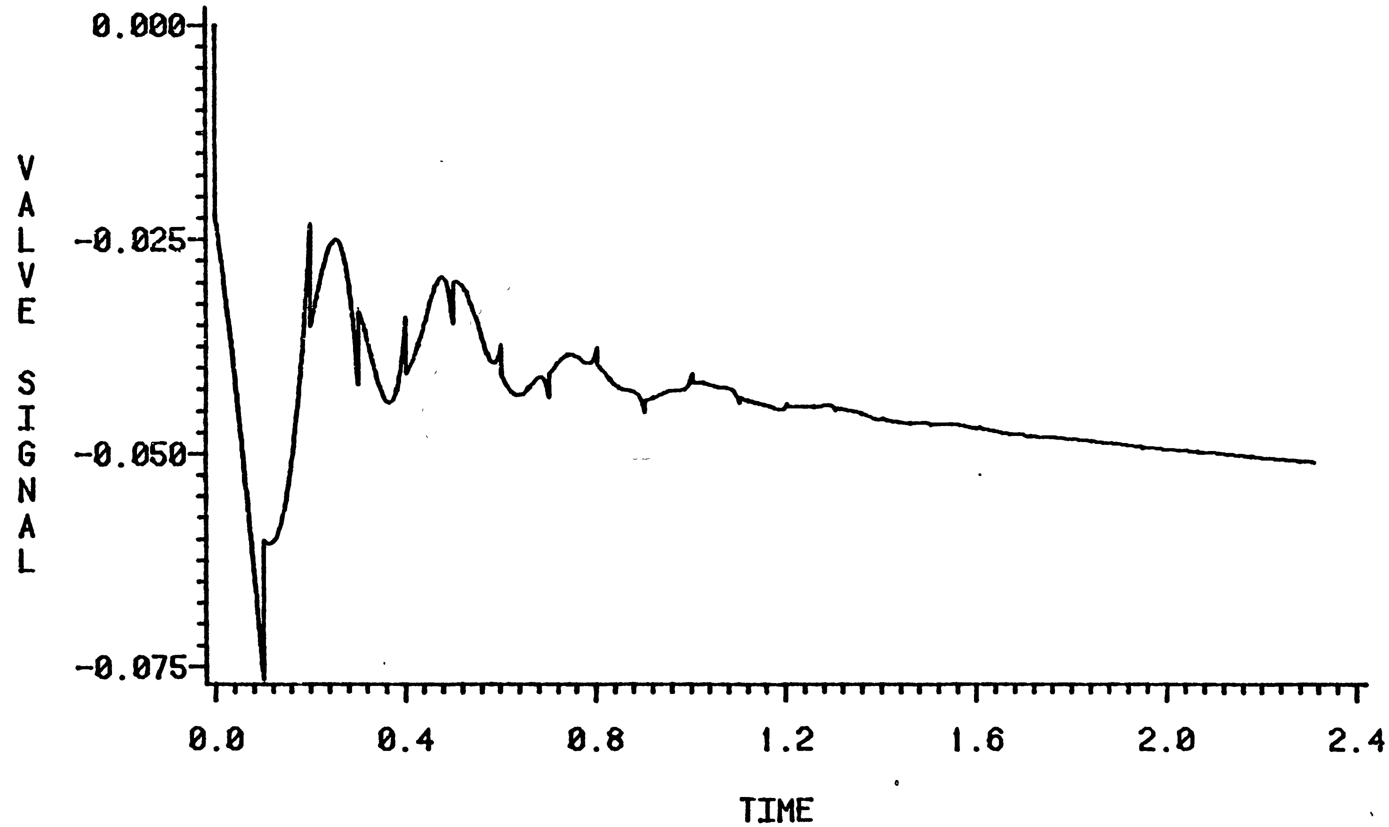


Figure 94. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 2.0$

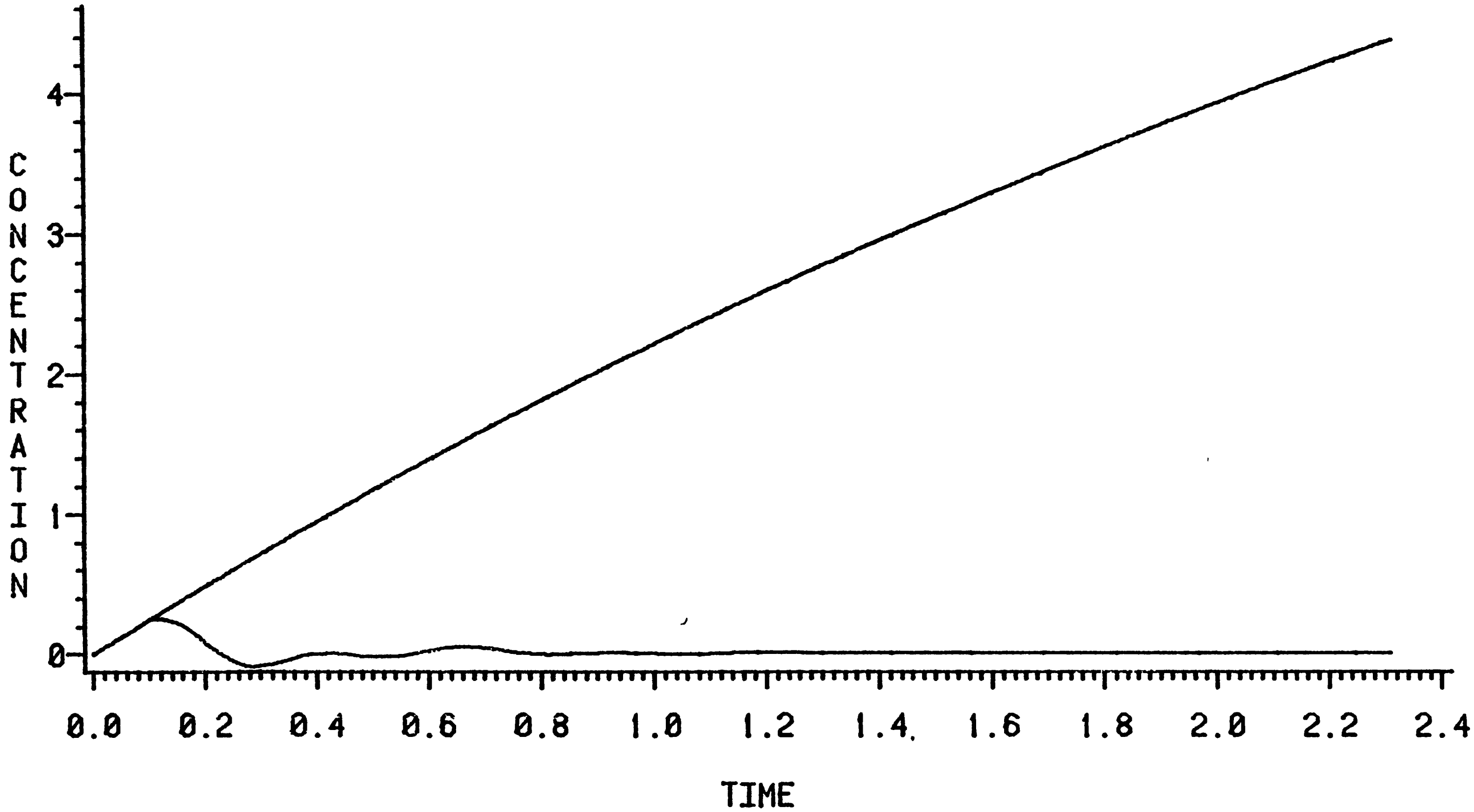


Figure 95. Concentration Variables, PID Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 4.0

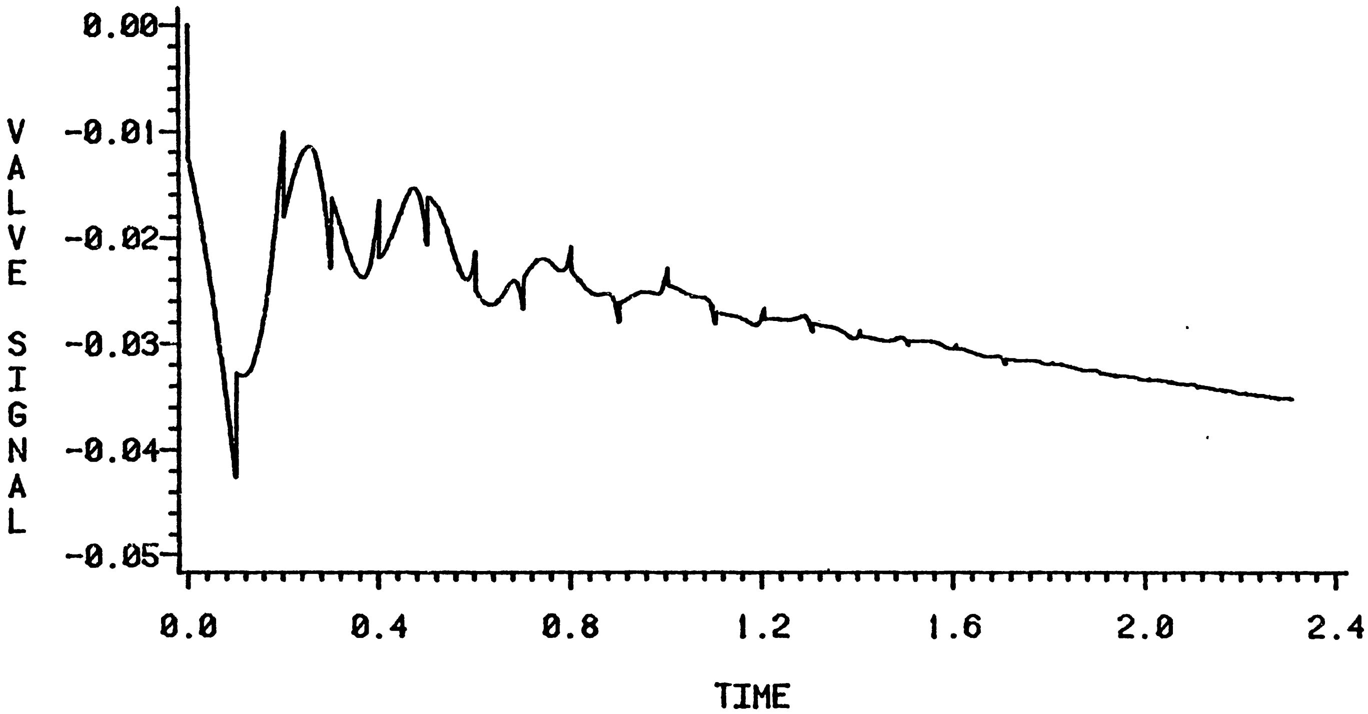


Figure 96. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 4.0$

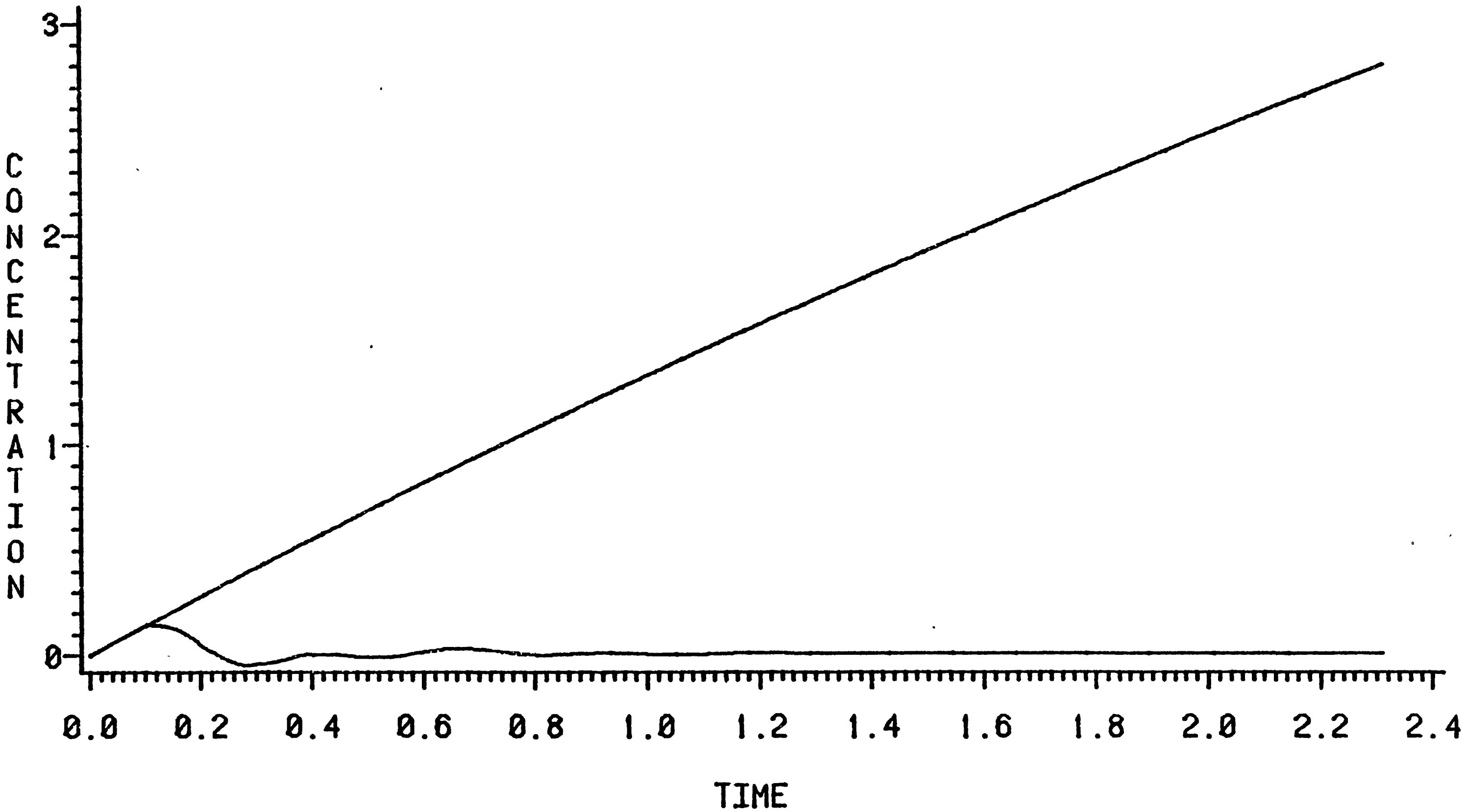


Figure 97. Concentration Variables, PID Controller, θ_d/τ_1 = 0.1, τ_2/τ_1 = 7.0

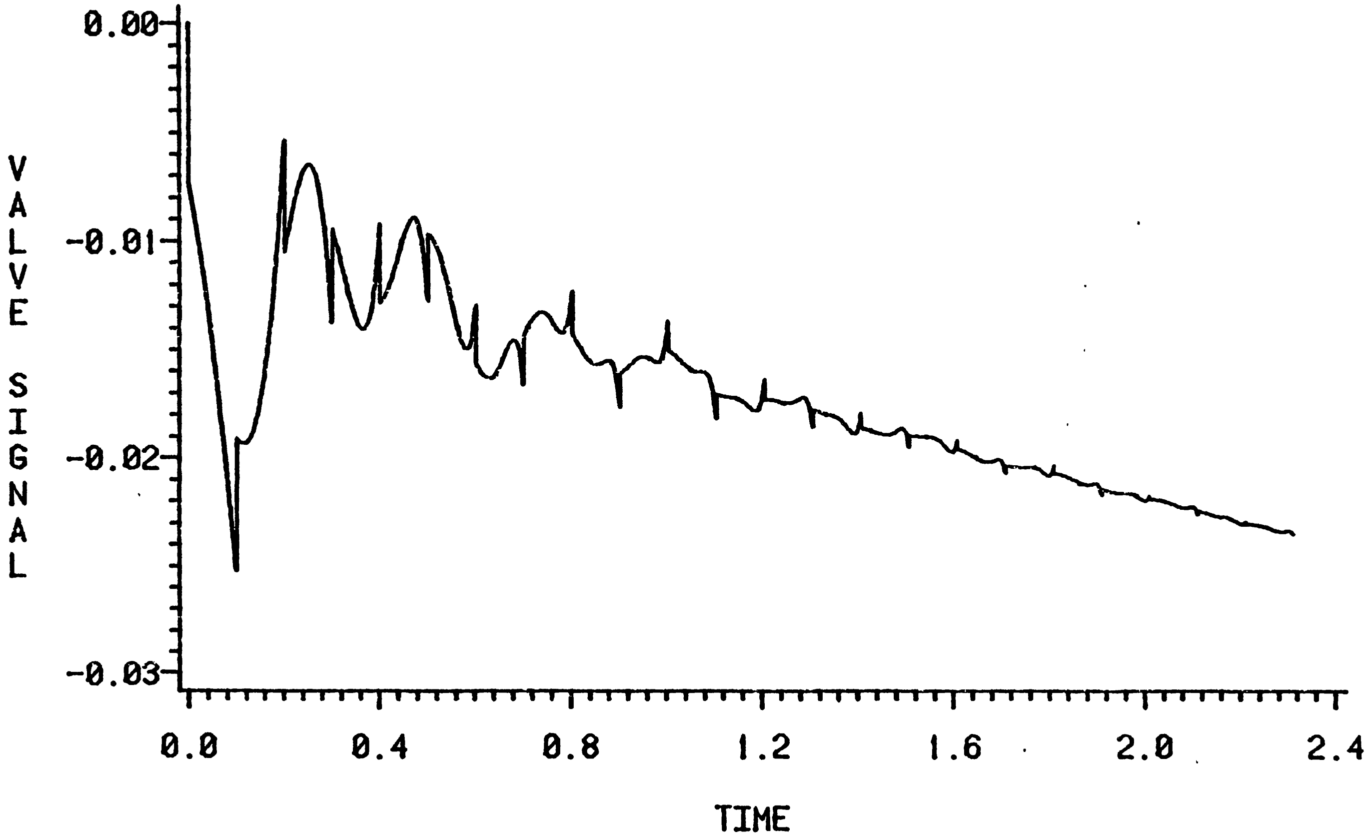


Figure 98. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.1$, $\tau_2/\tau_1 = 7.0$

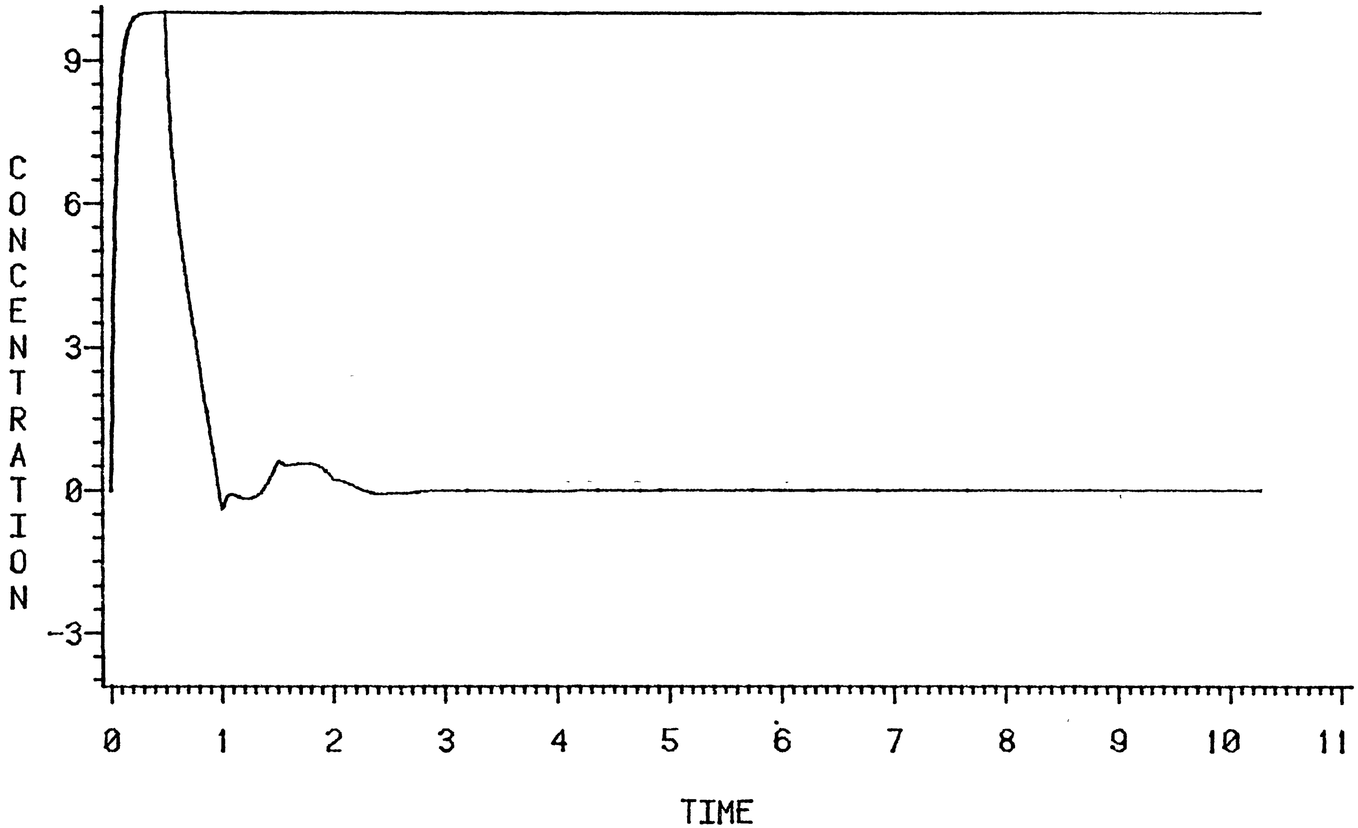


Figure 99. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.05$

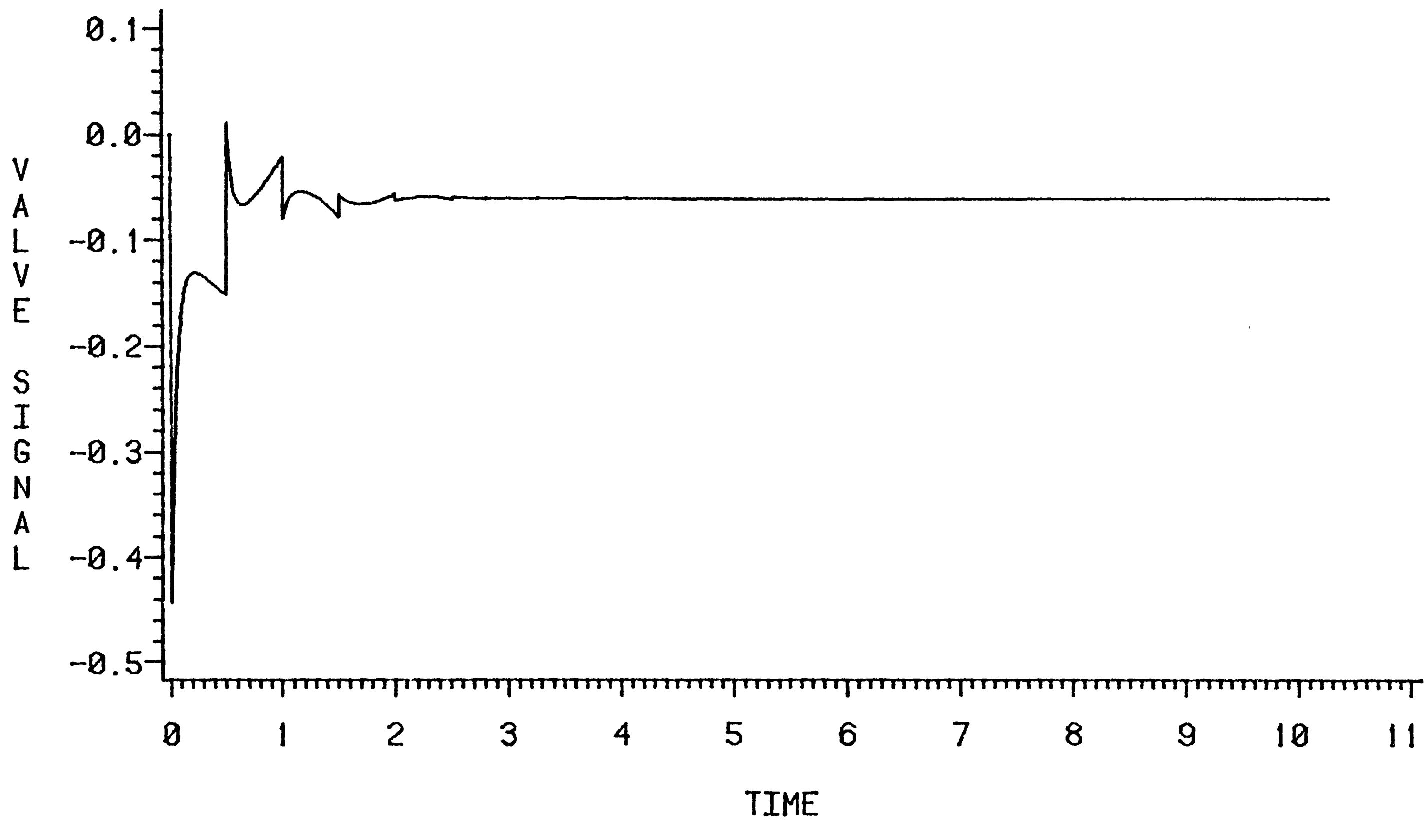


Figure 100. Valve Signal, PID Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 0.05

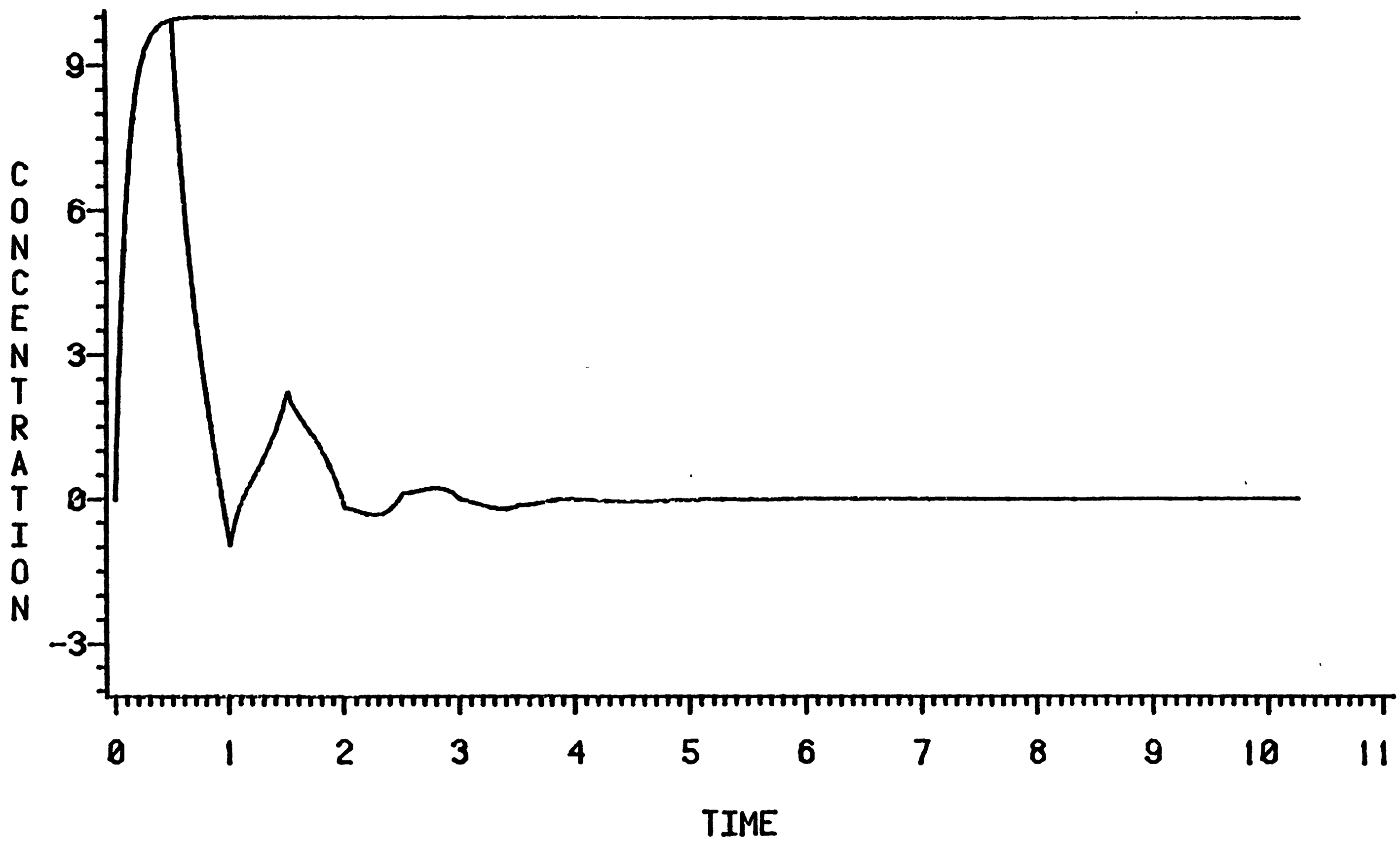


Figure 101. Concentration Variables, PID Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 0.1

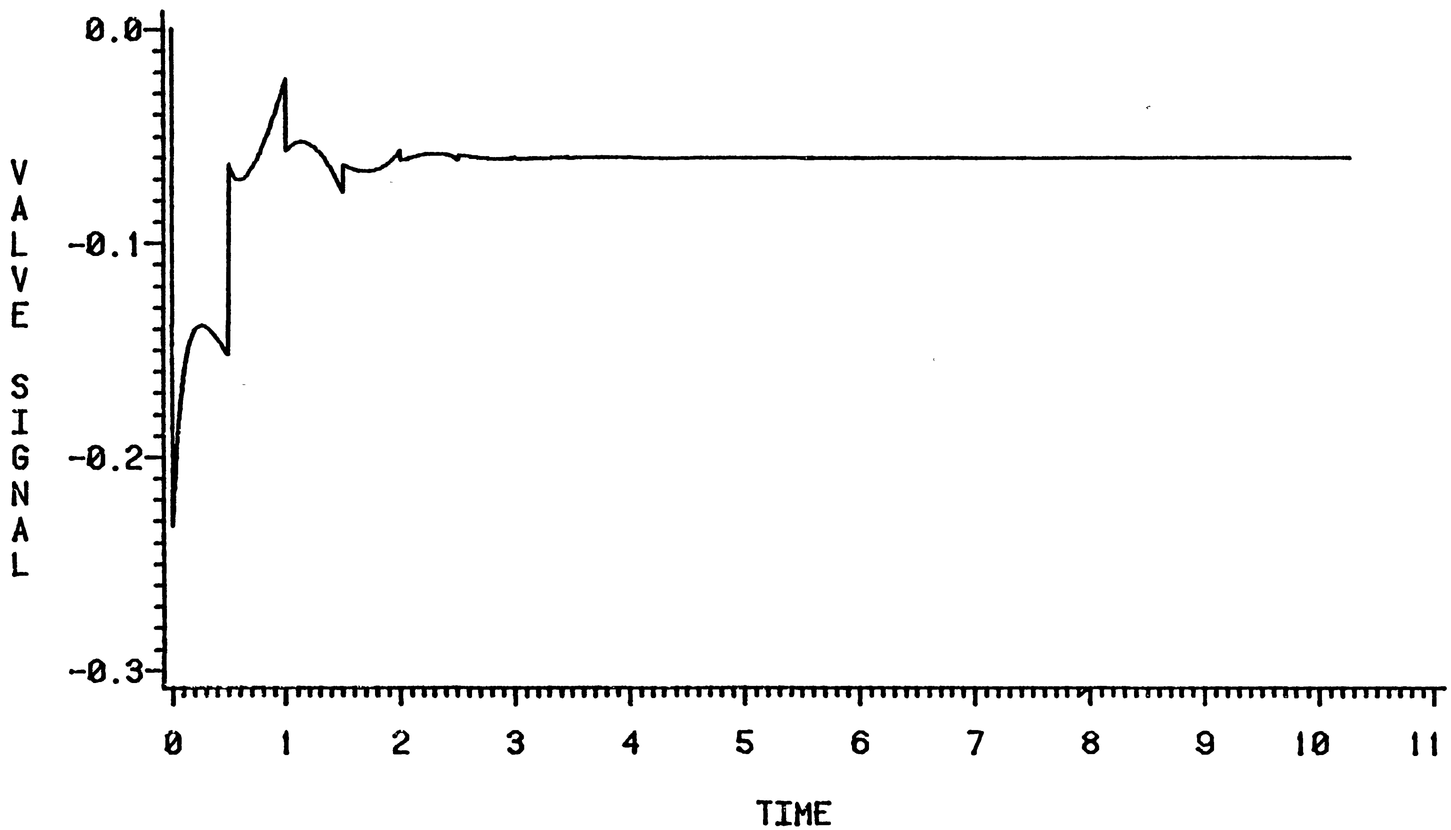


Figure 102. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.1$

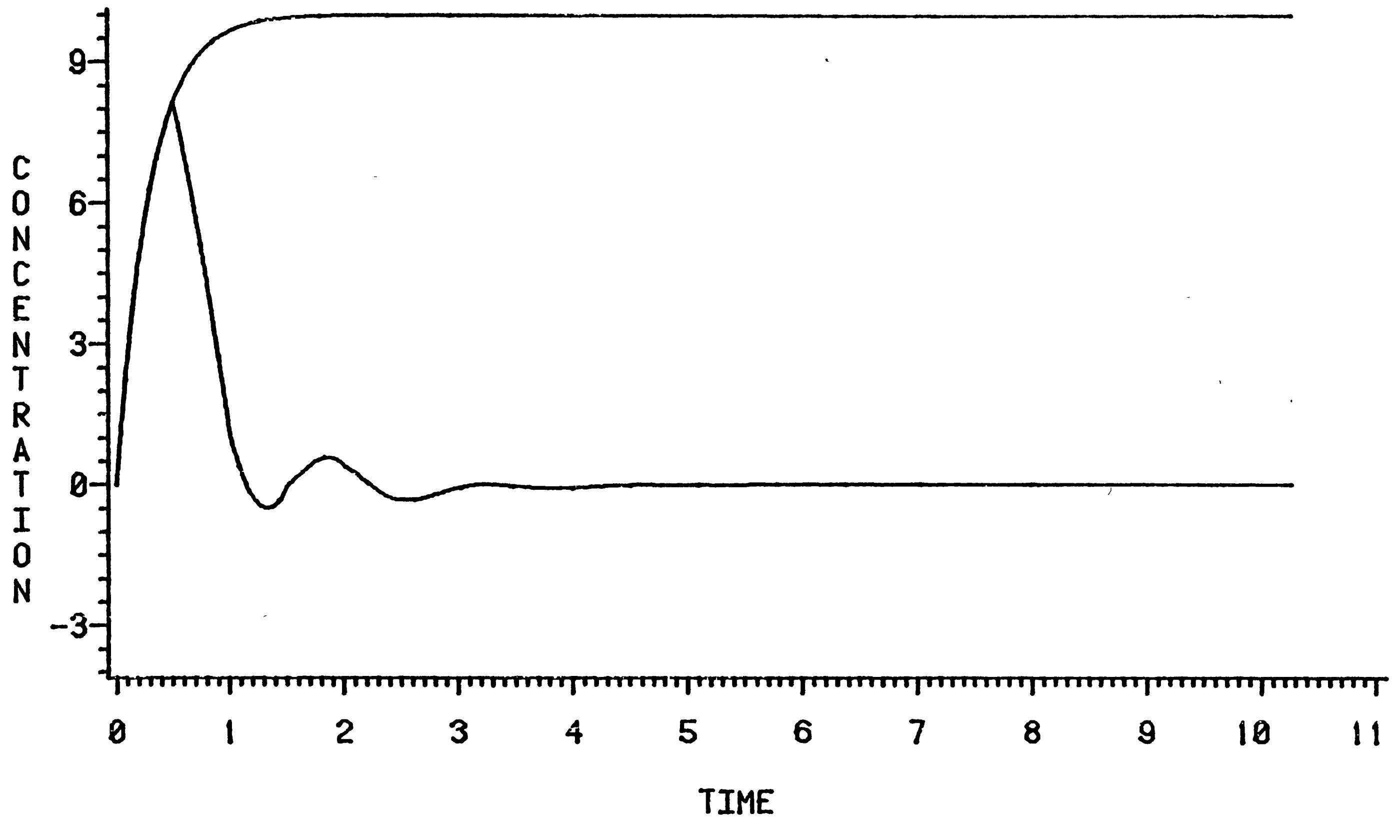


Figure 103. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.3$

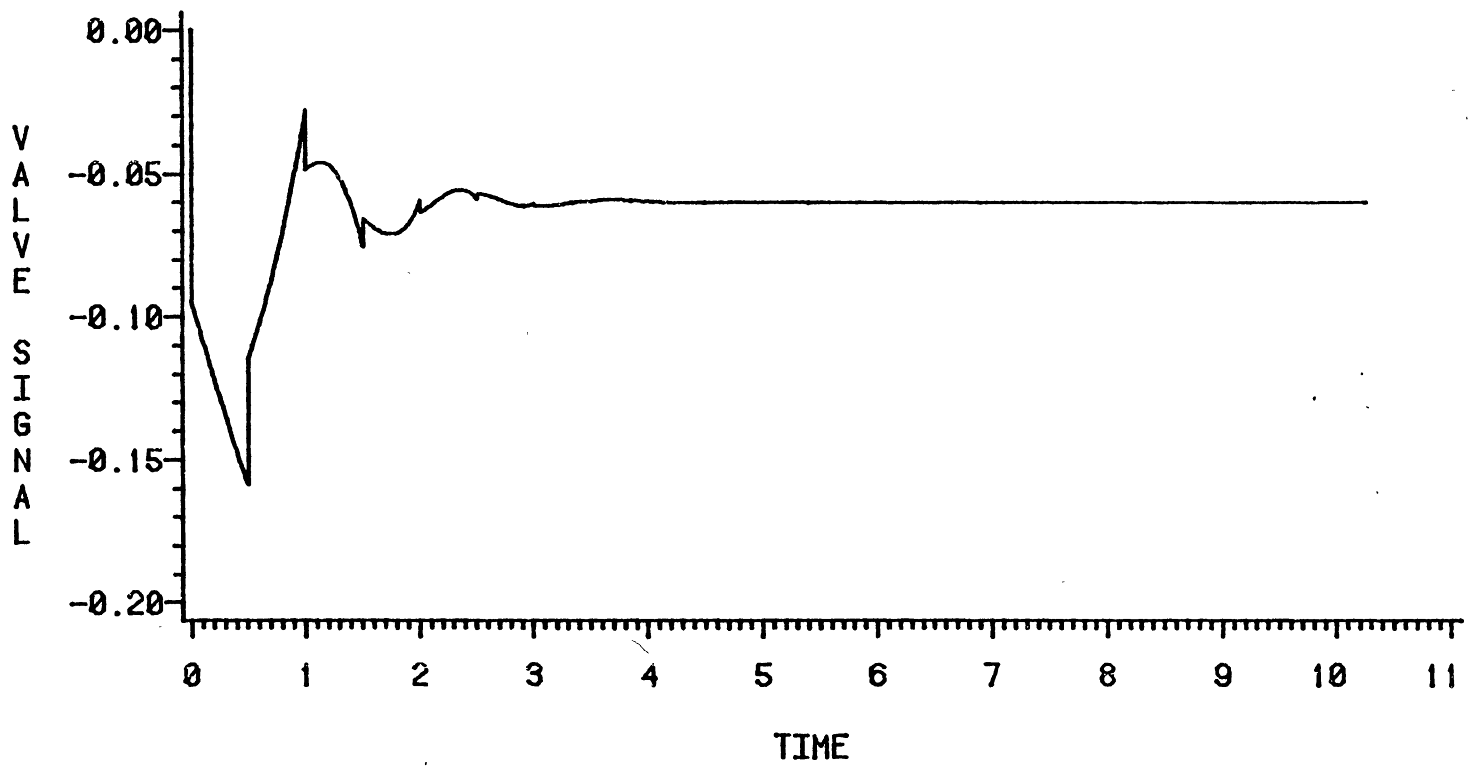


Figure 104. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.3$

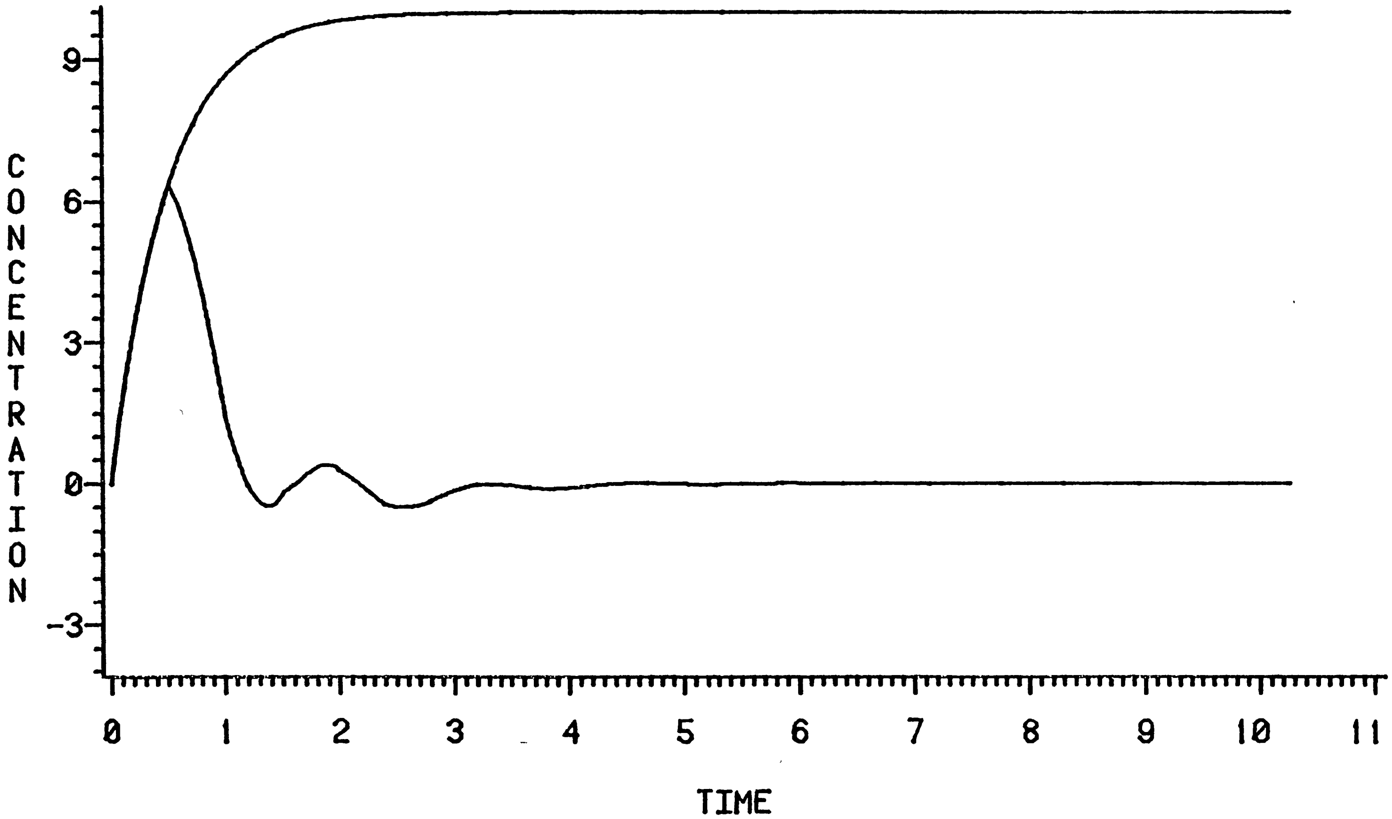


Figure 105. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.5$

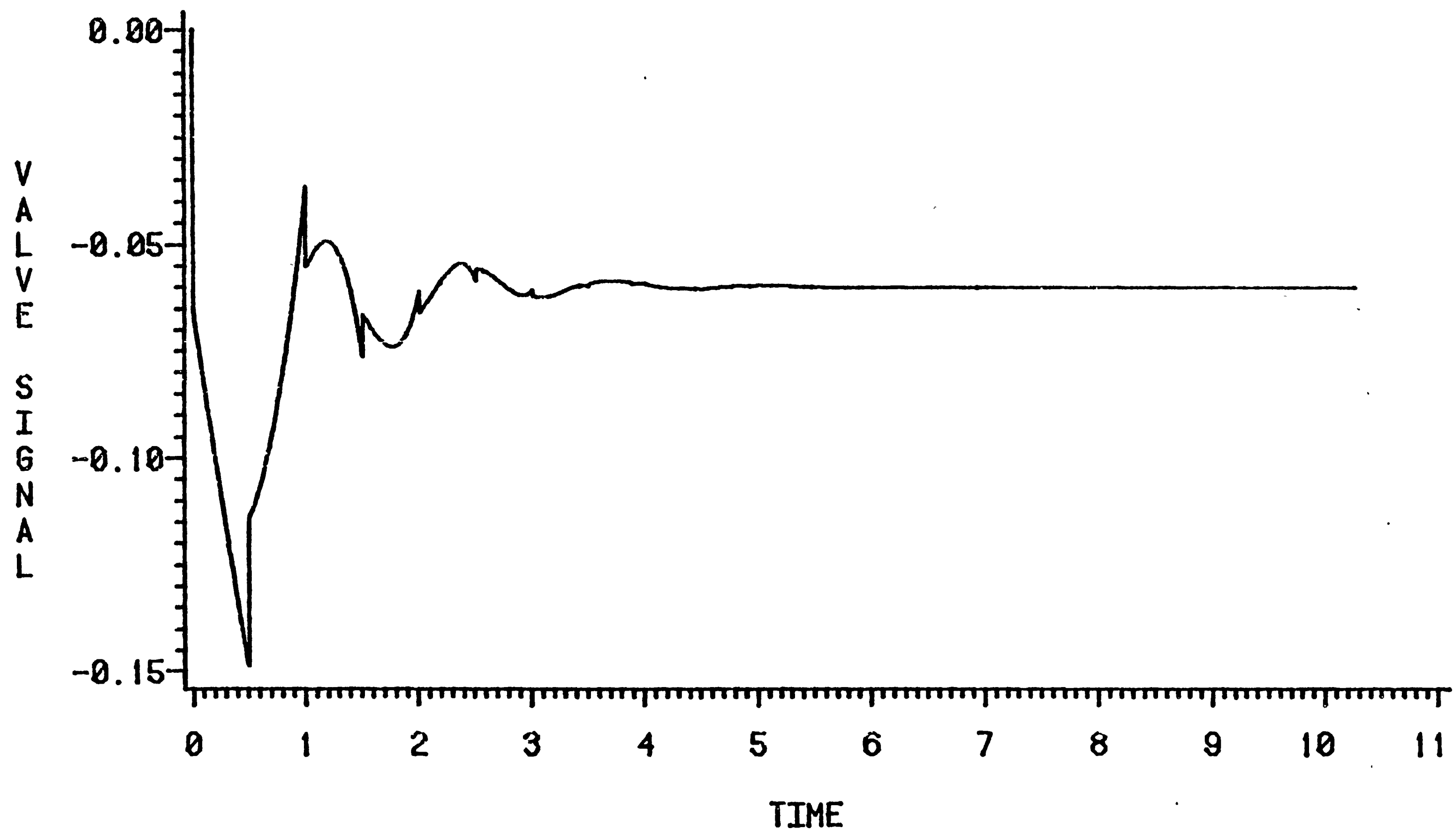


Figure 106. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 0.5$

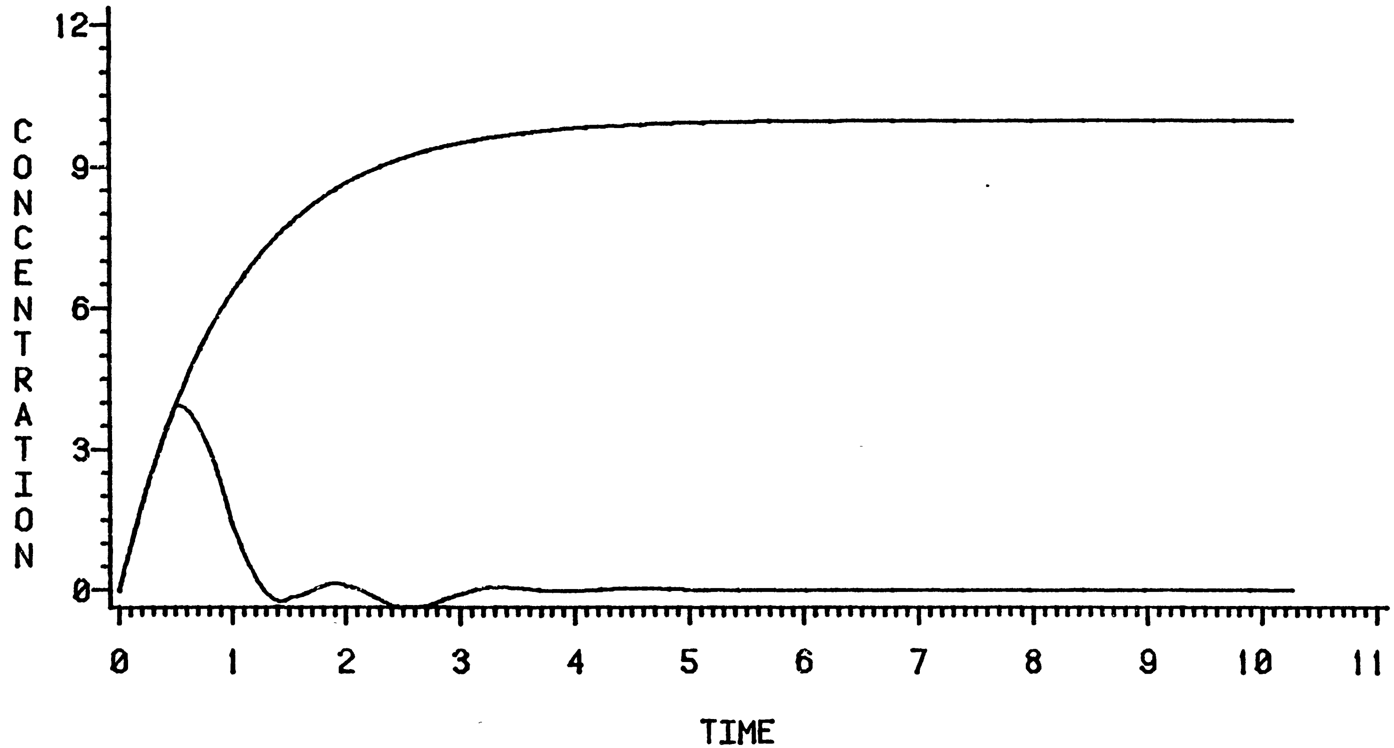


Figure 107. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 1.0$

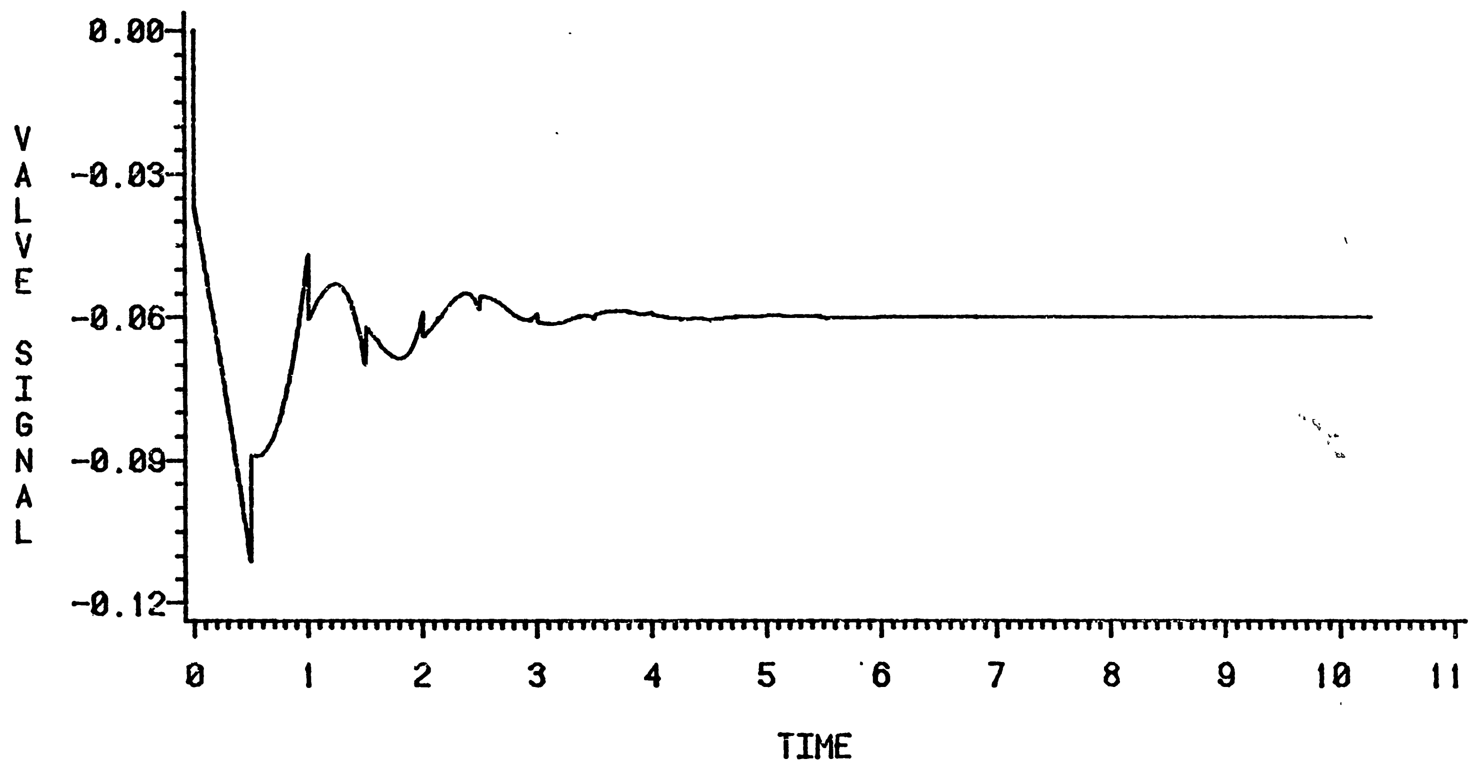


Figure 108. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 1.0$

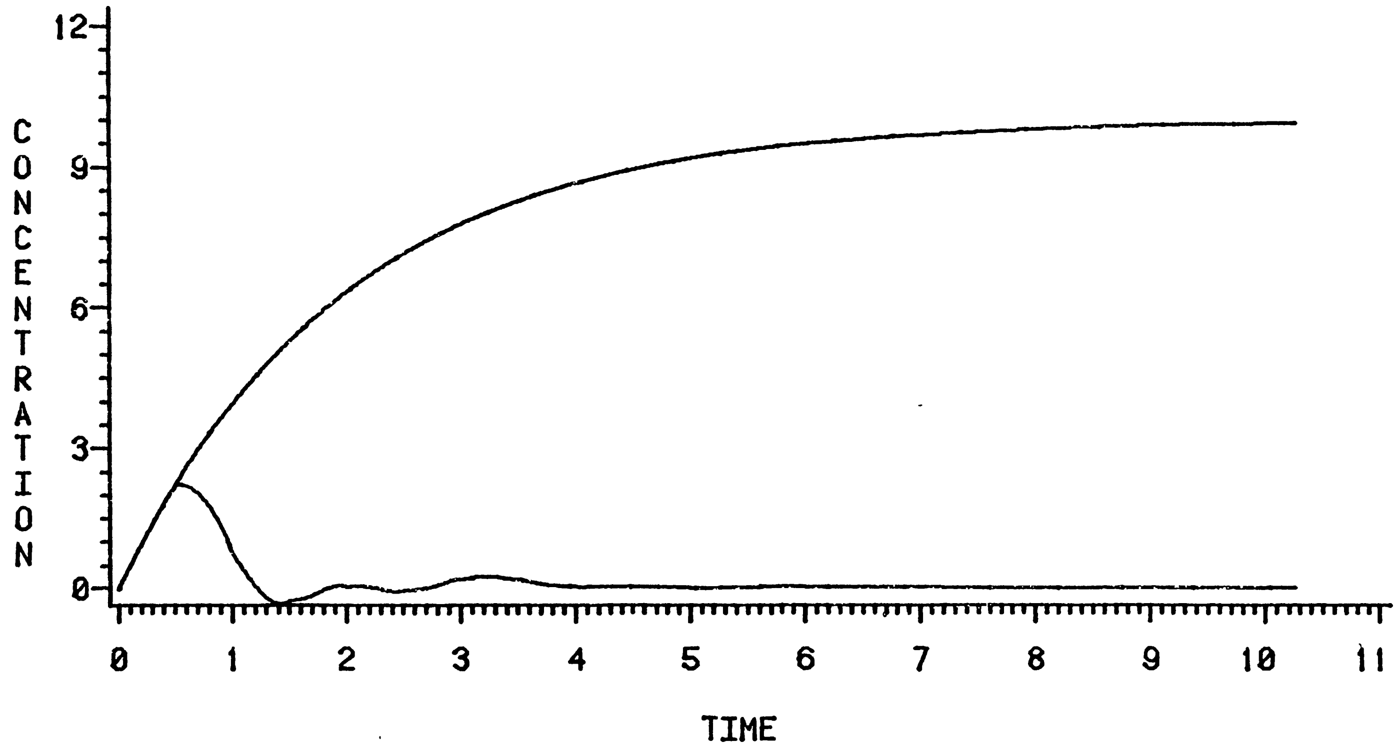


Figure 109. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 2.0$

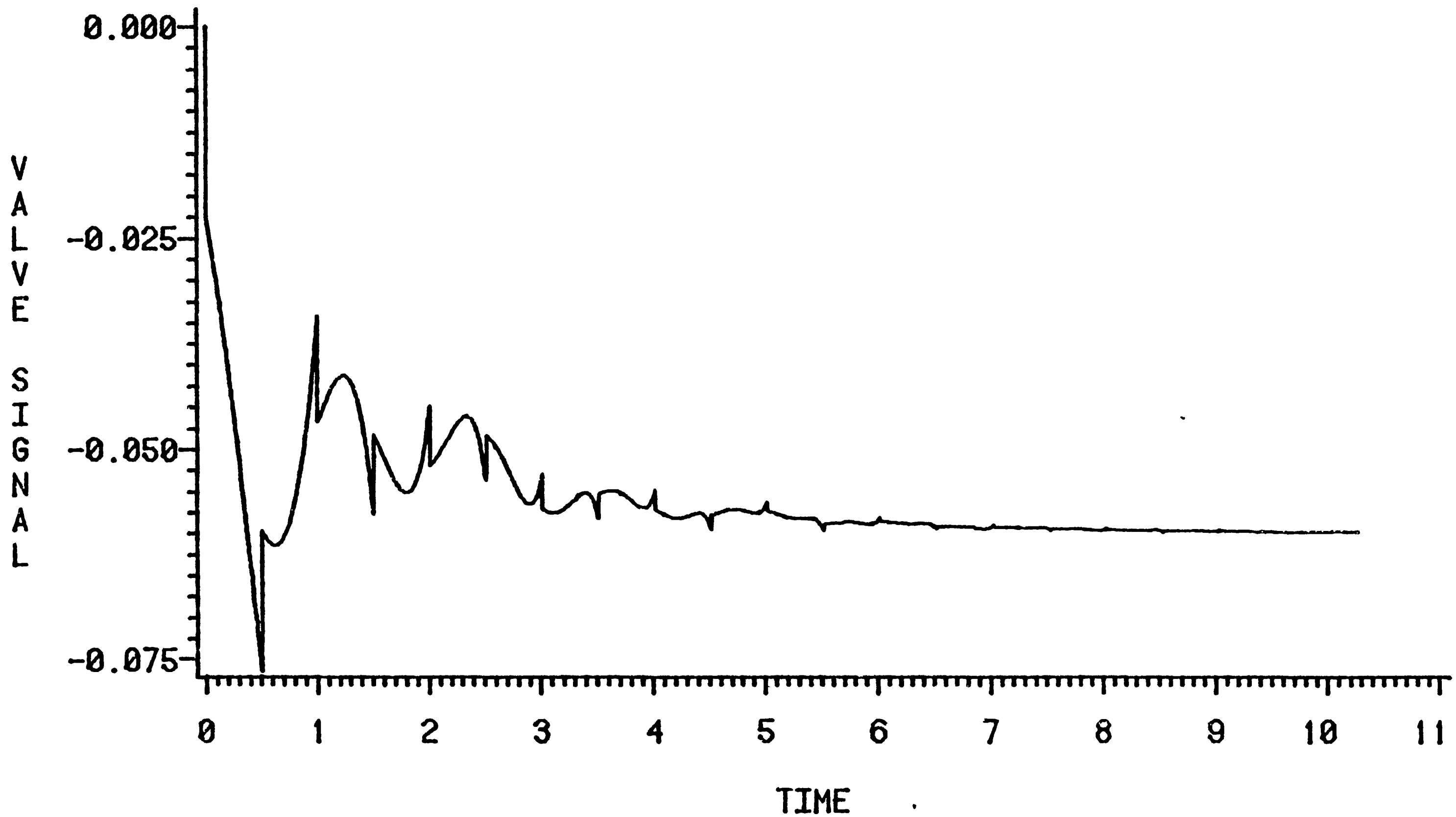


Figure 110. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 2.0$

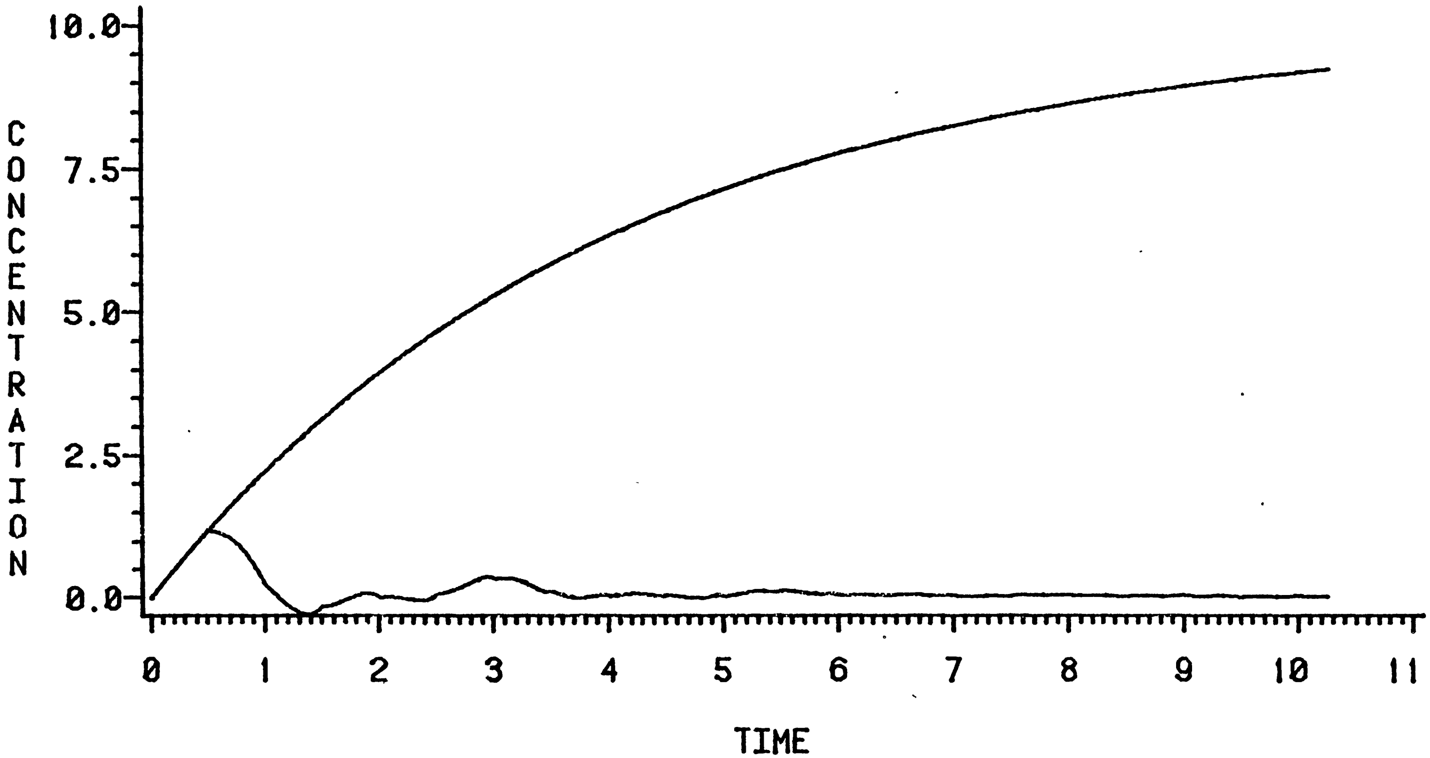


Figure 111. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 4.0$

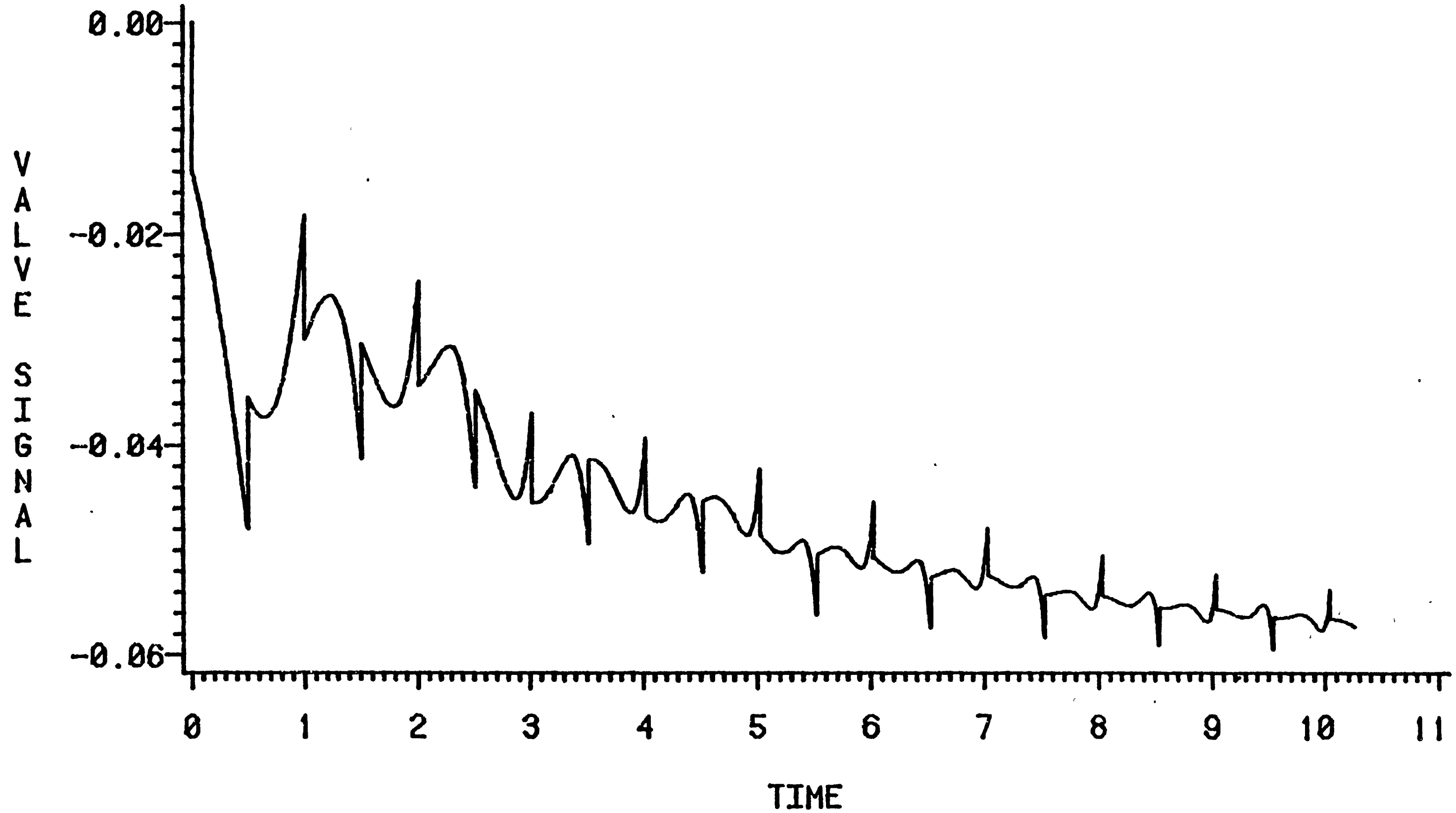


Figure 112. Valve Signal, PID Controller, θ_d/τ_1 = 0.5, τ_2/τ_1 = 4.0

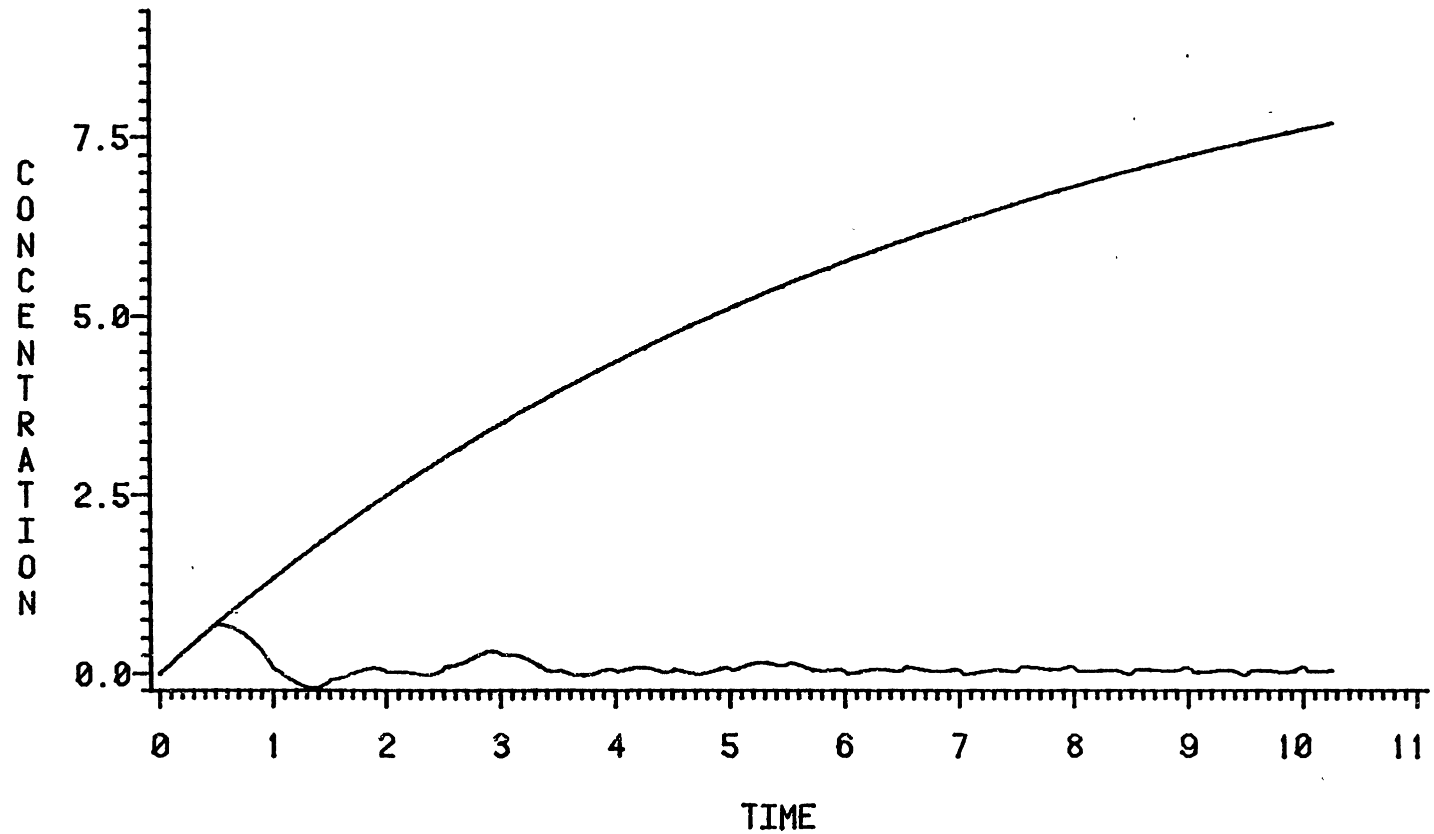


Figure 113. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 7.0$

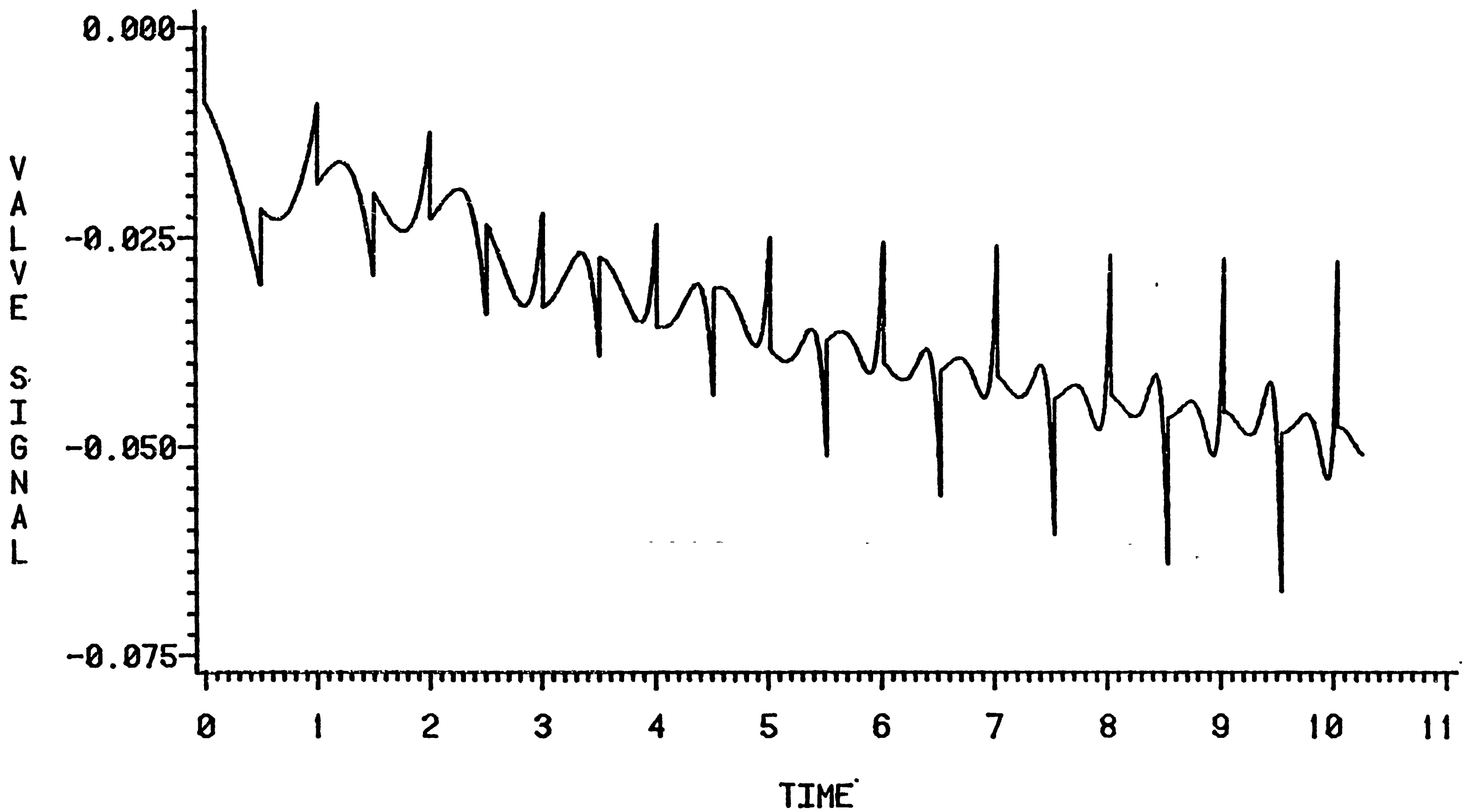


Figure 114. Valve Signal, PID Controller, $\theta_d/\tau_1 = 0.5$, $\tau_2/\tau_1 = 7.0$

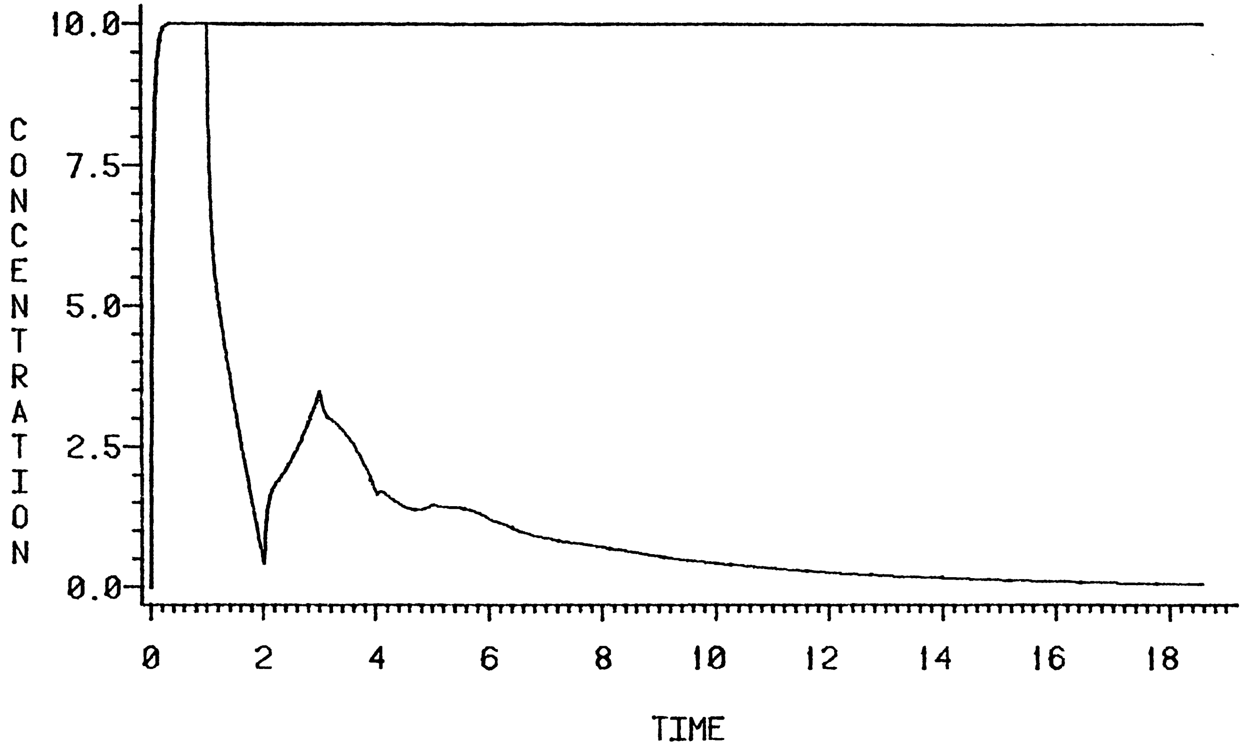


Figure 115. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.05$

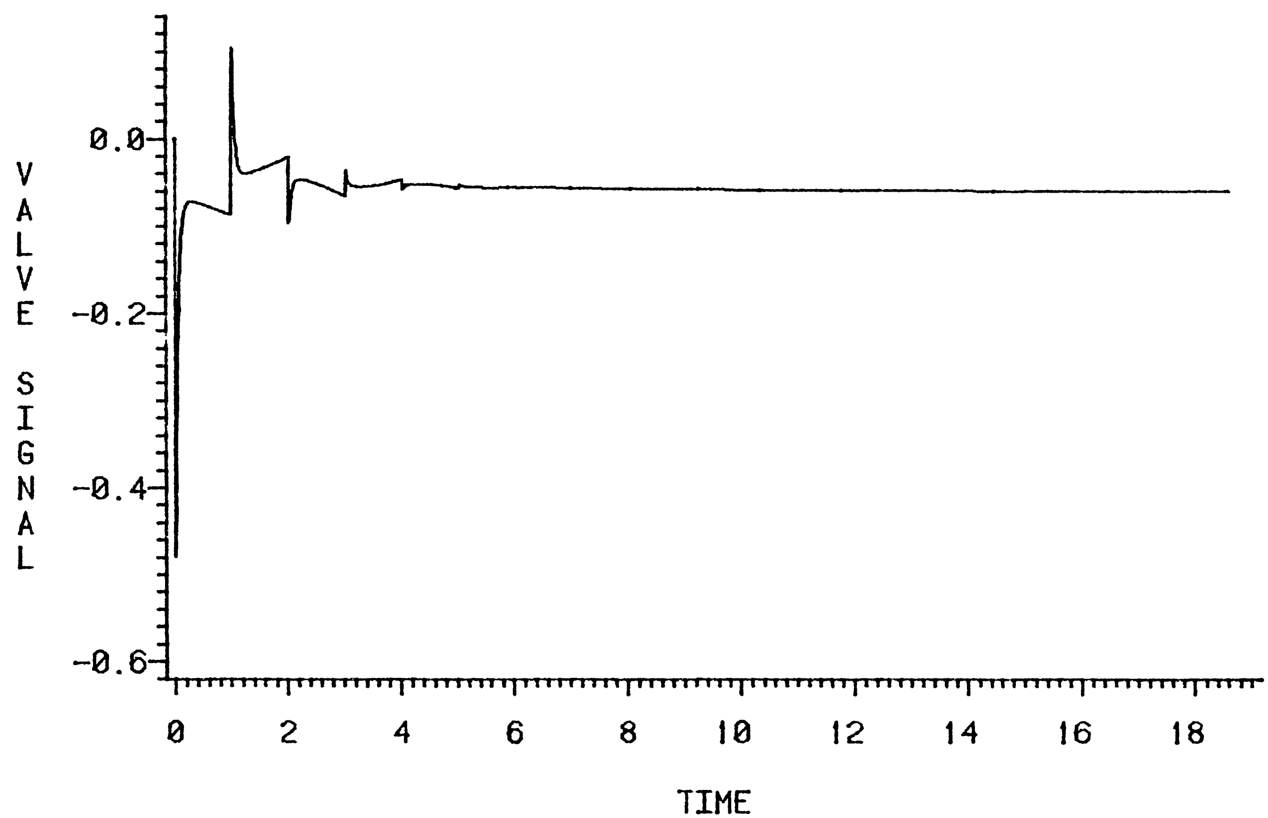


Figure 116. Valve Signal, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.05$

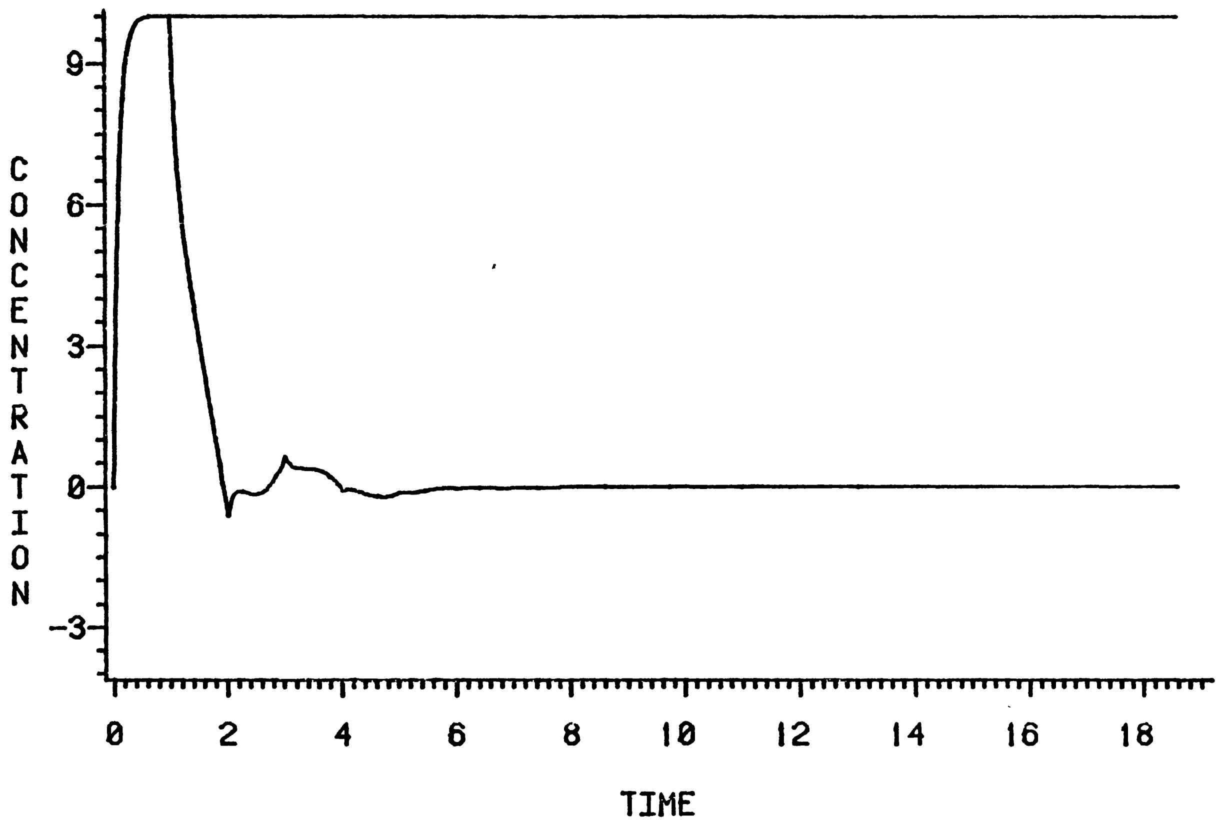


Figure 117. Concentration Variables, PID Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 0.1

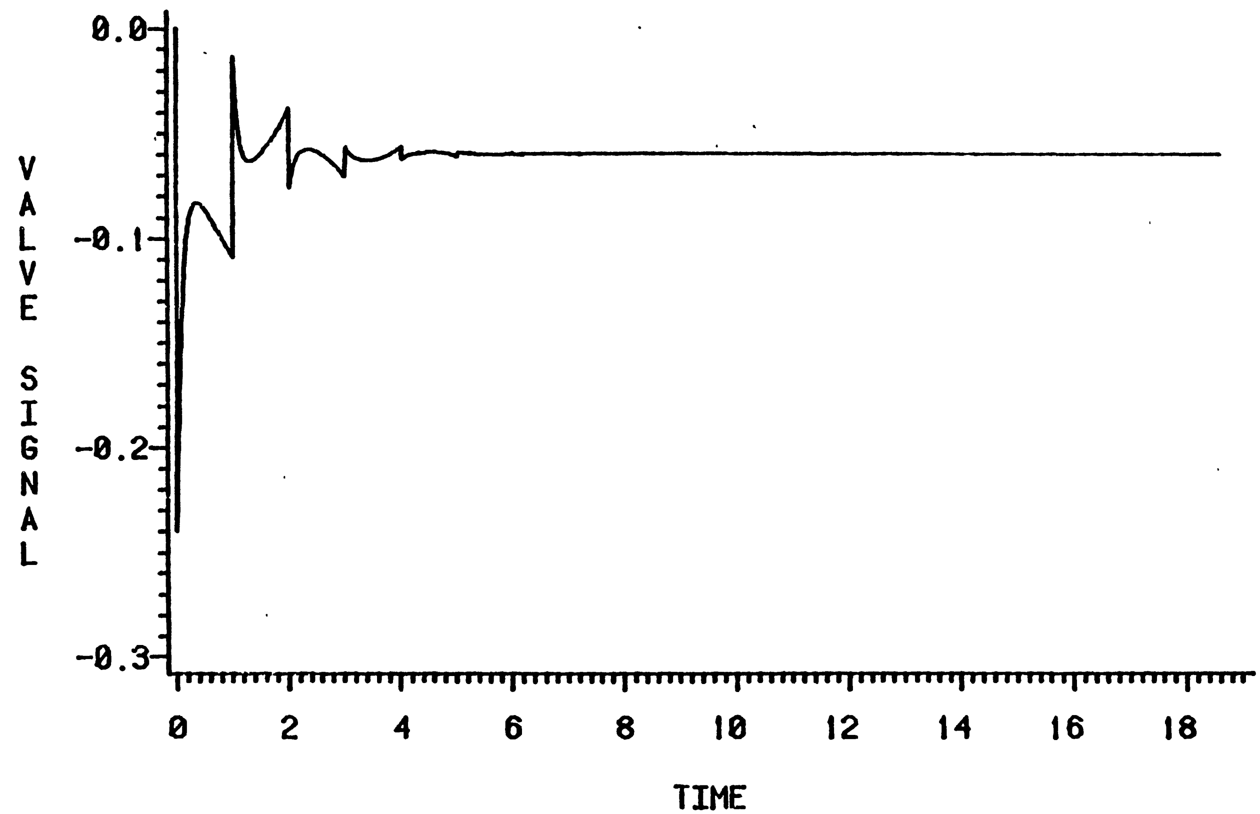


Figure 118. Valve Signal, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.1$

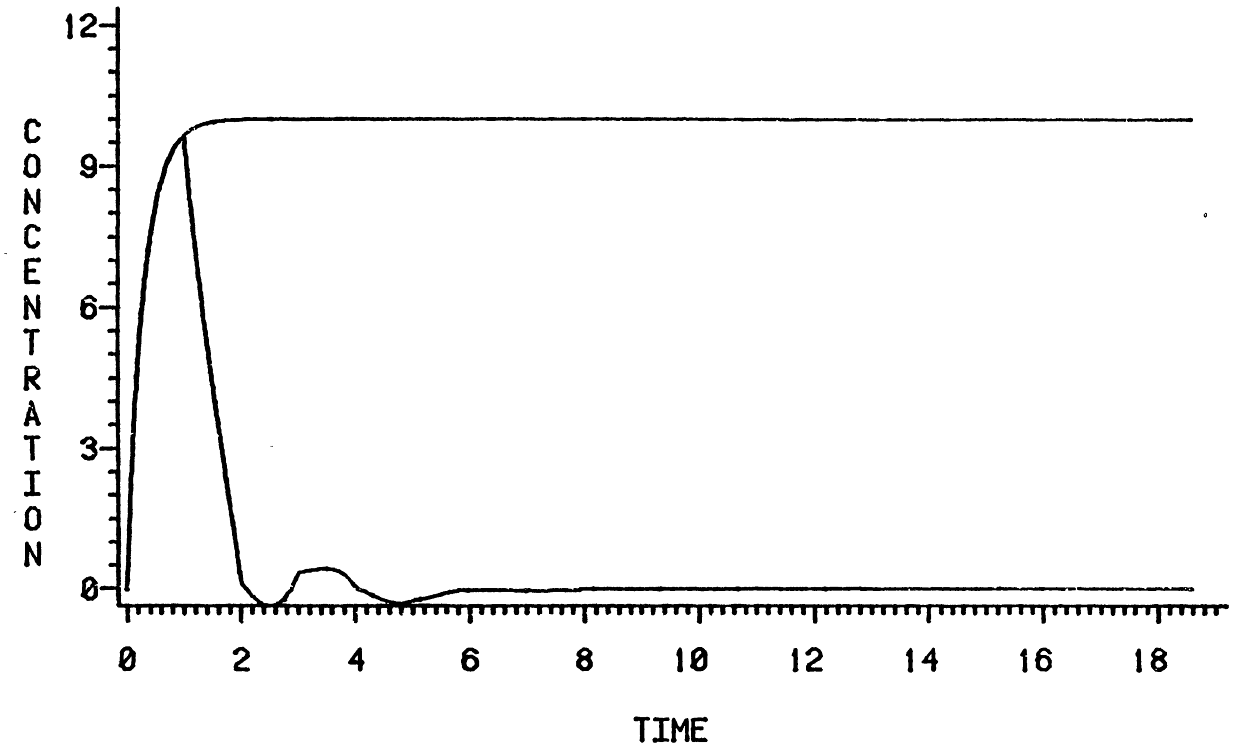


Figure 119. Concentration Variables, PID Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 0.3

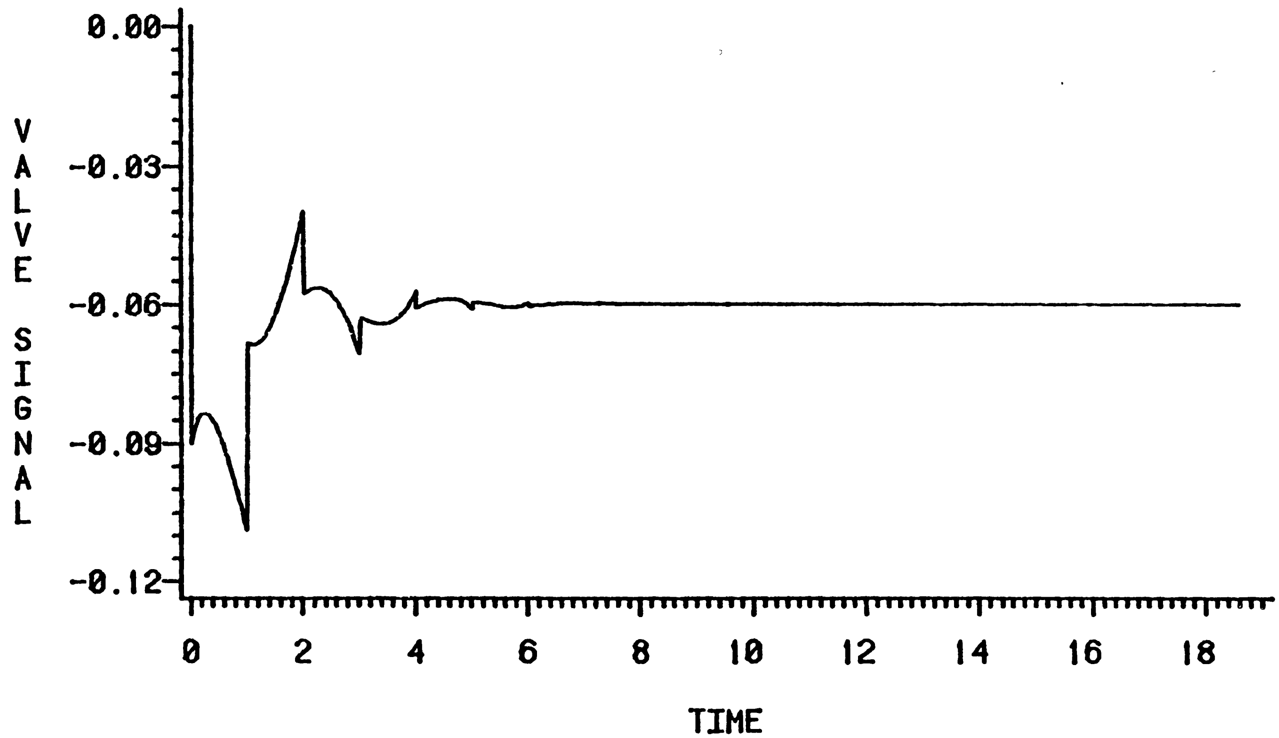


Figure 120. Valve Signal, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.3$

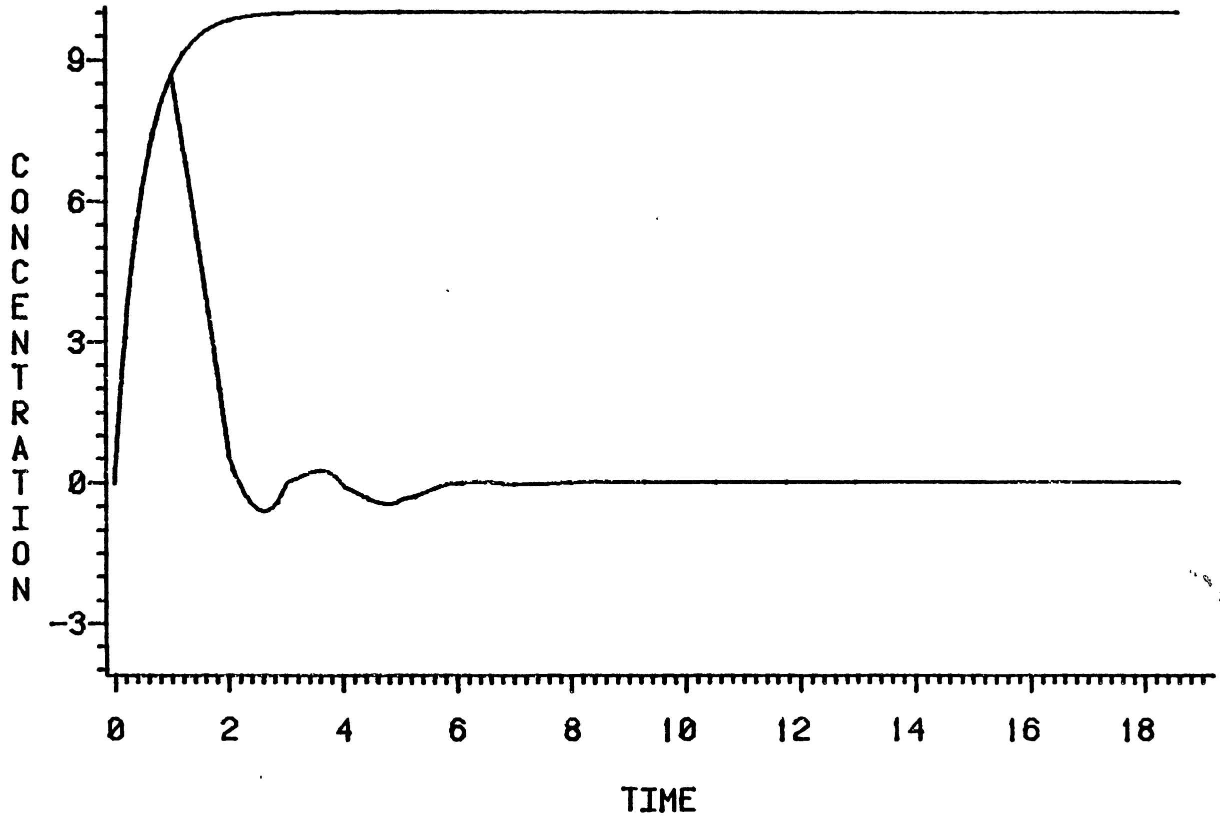


Figure 121. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 0.5$

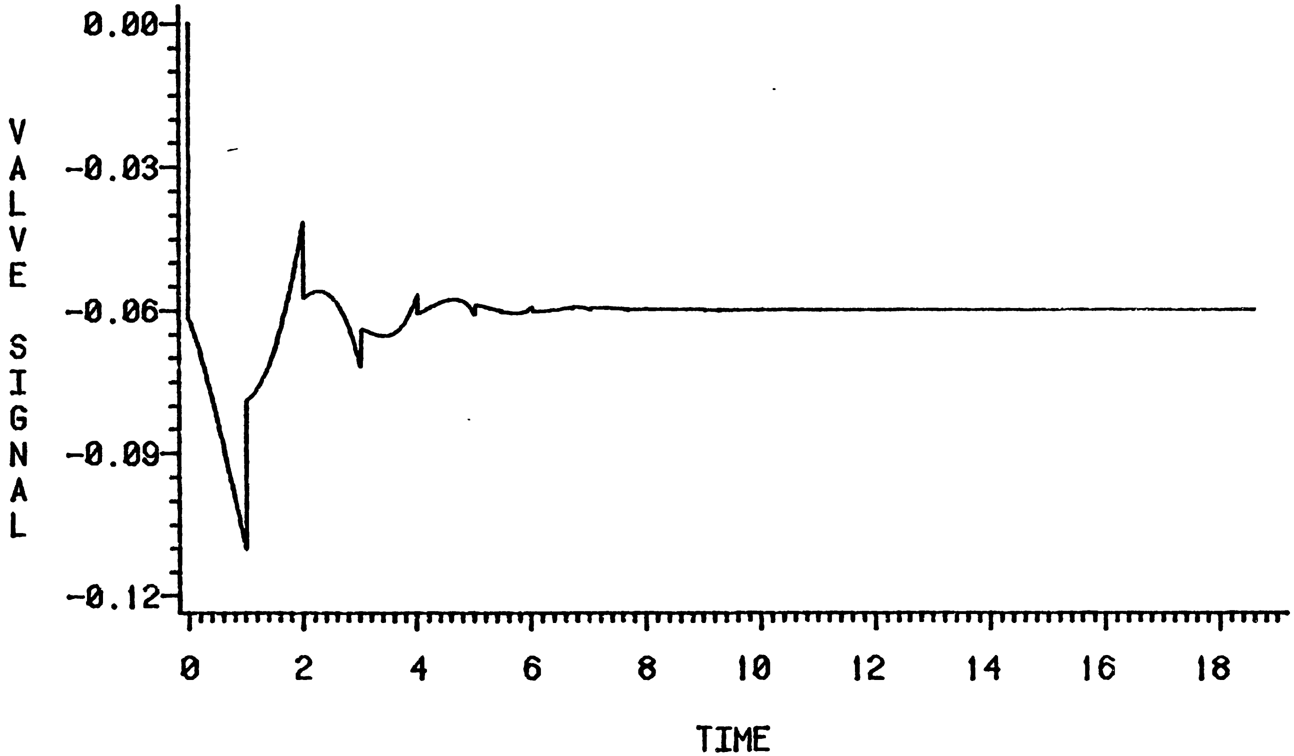


Figure 122. Valve Signal, PID Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 0.5

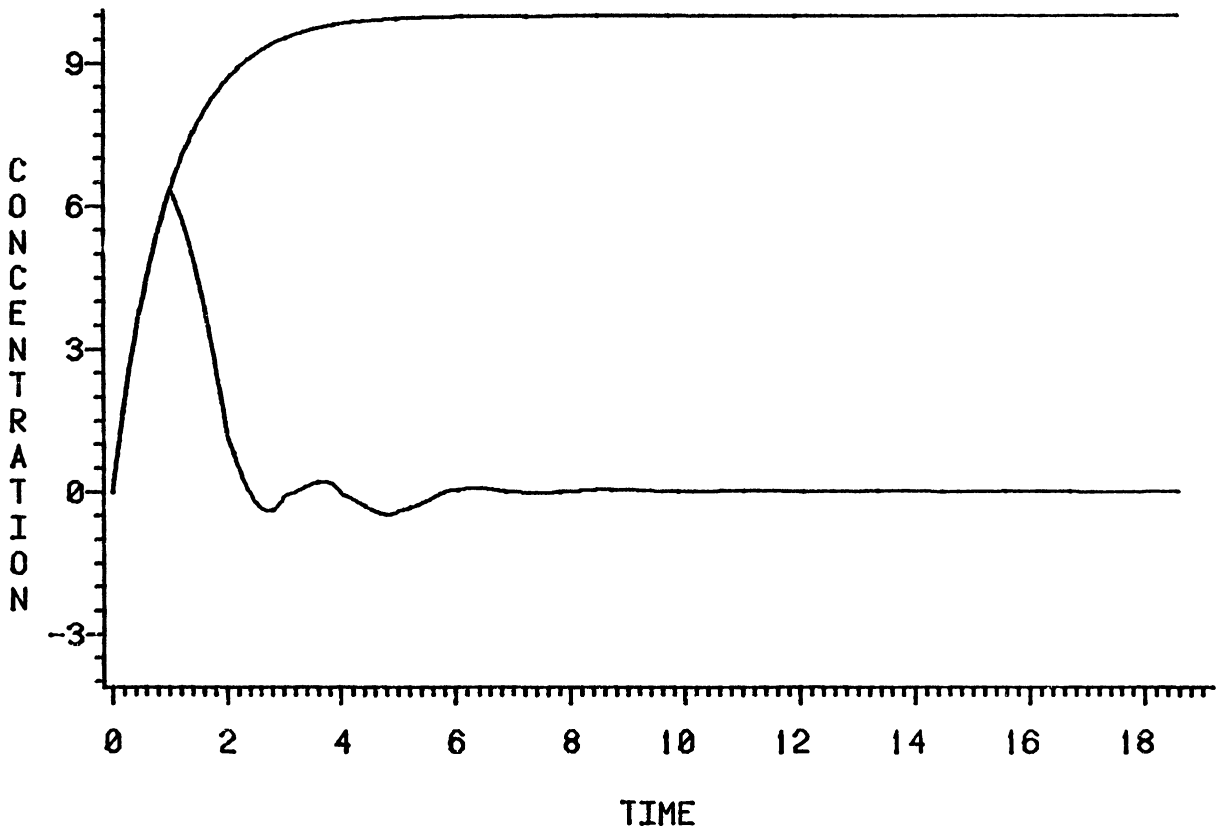


Figure 123. Concentration Variables, PID Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 1.0

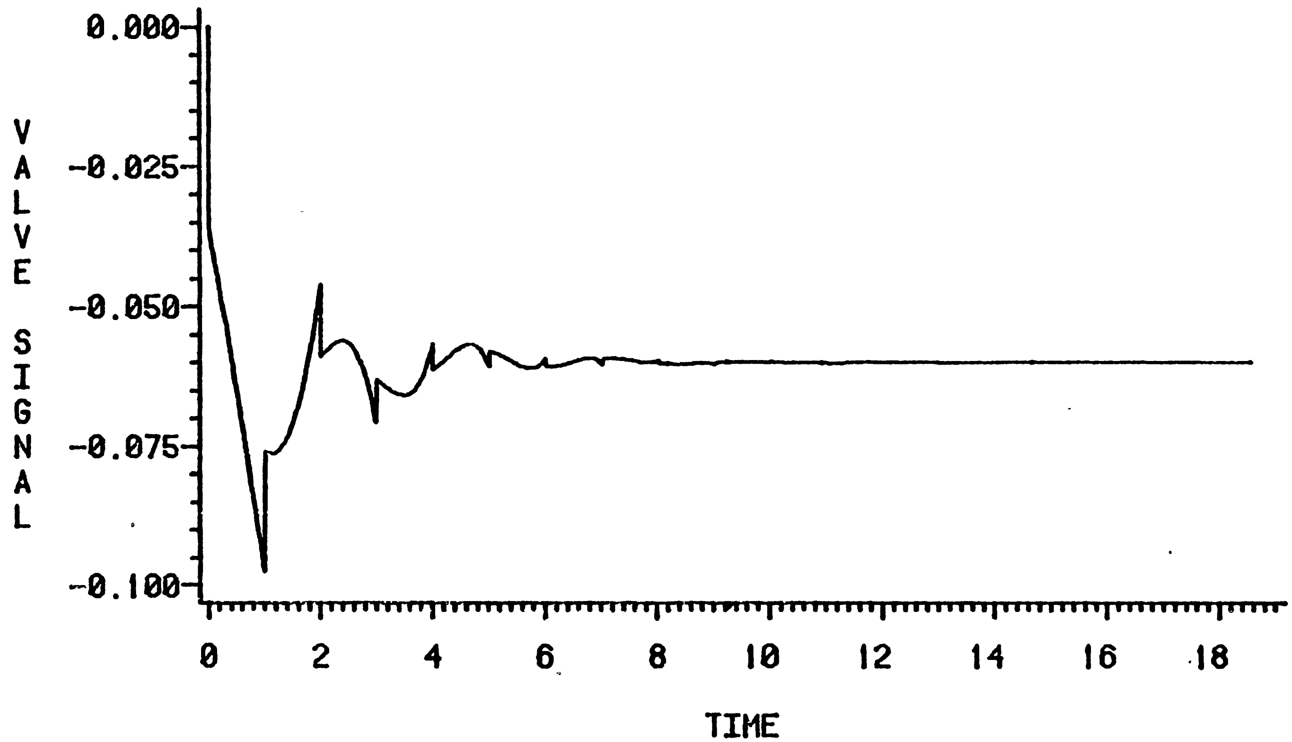


Figure 124. Valve Signal, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 1.0$

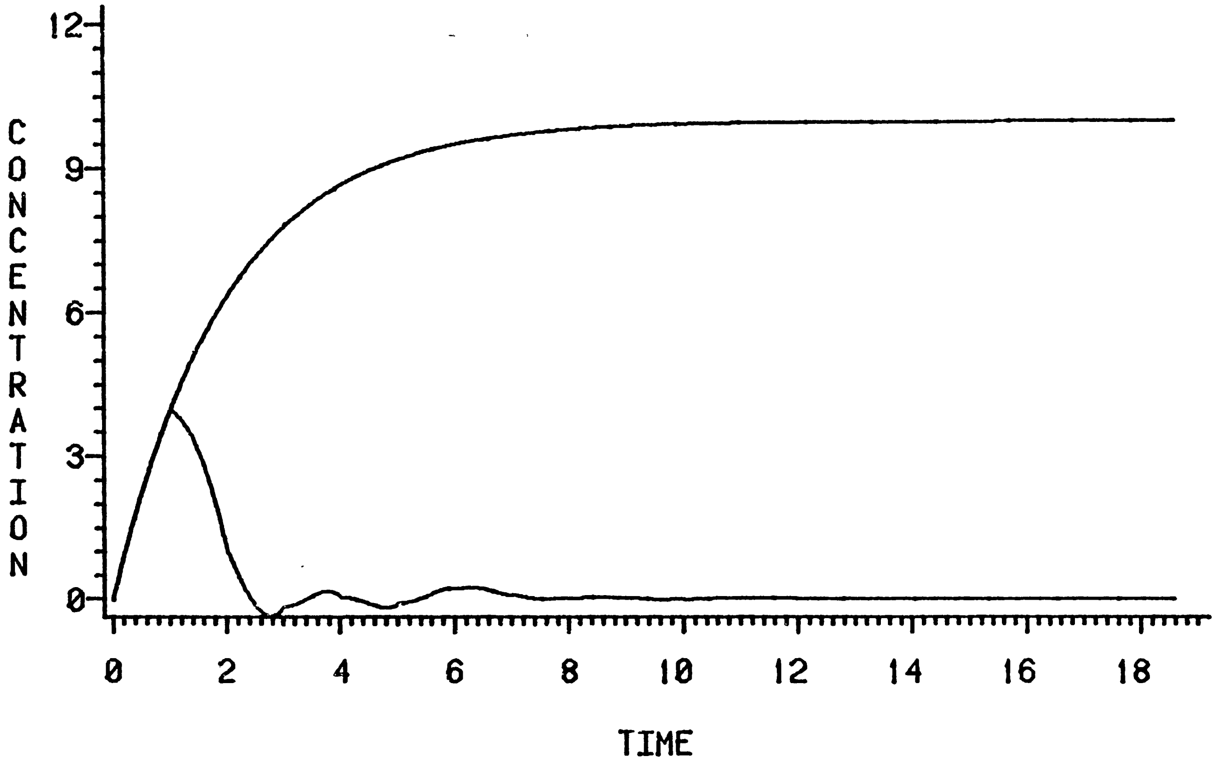


Figure 125. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 2.0$

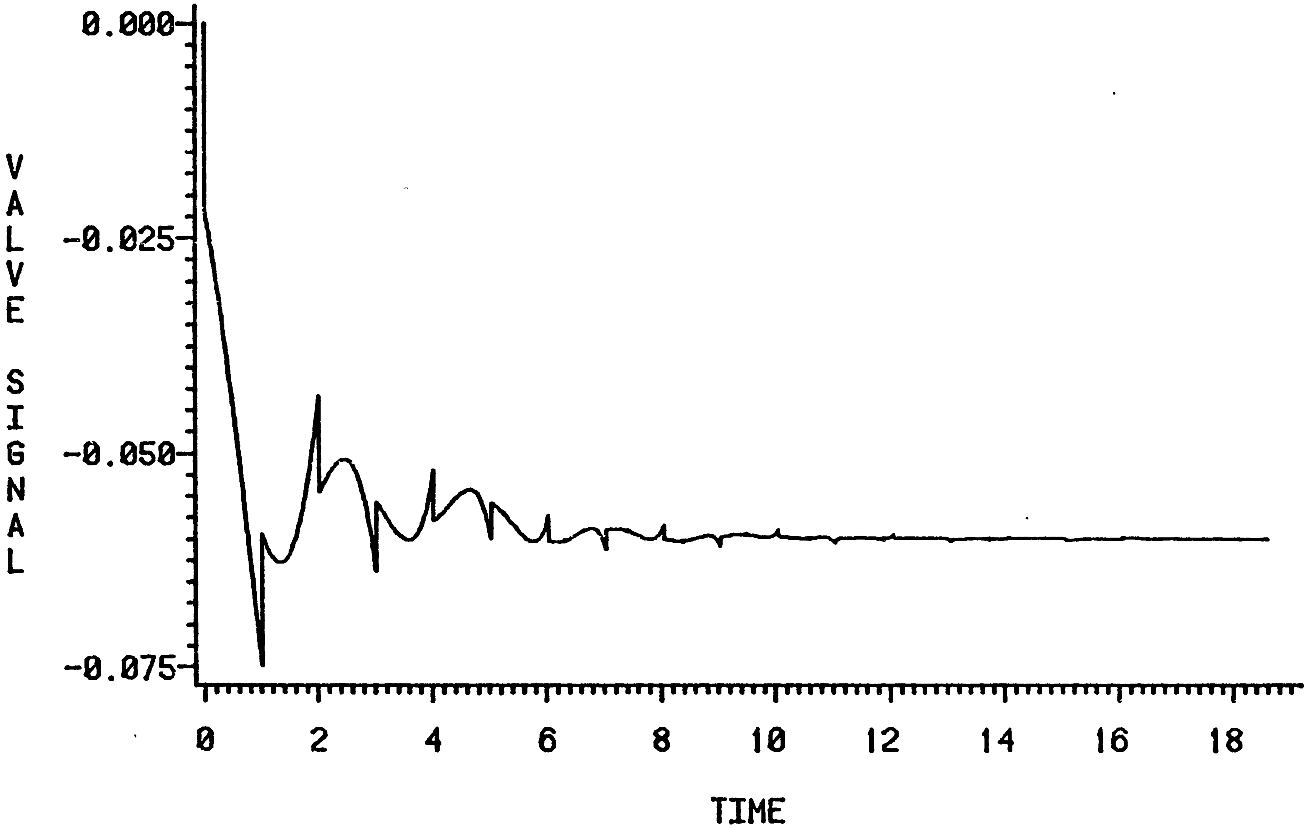


Figure 126. Valve Signal, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 2.0$

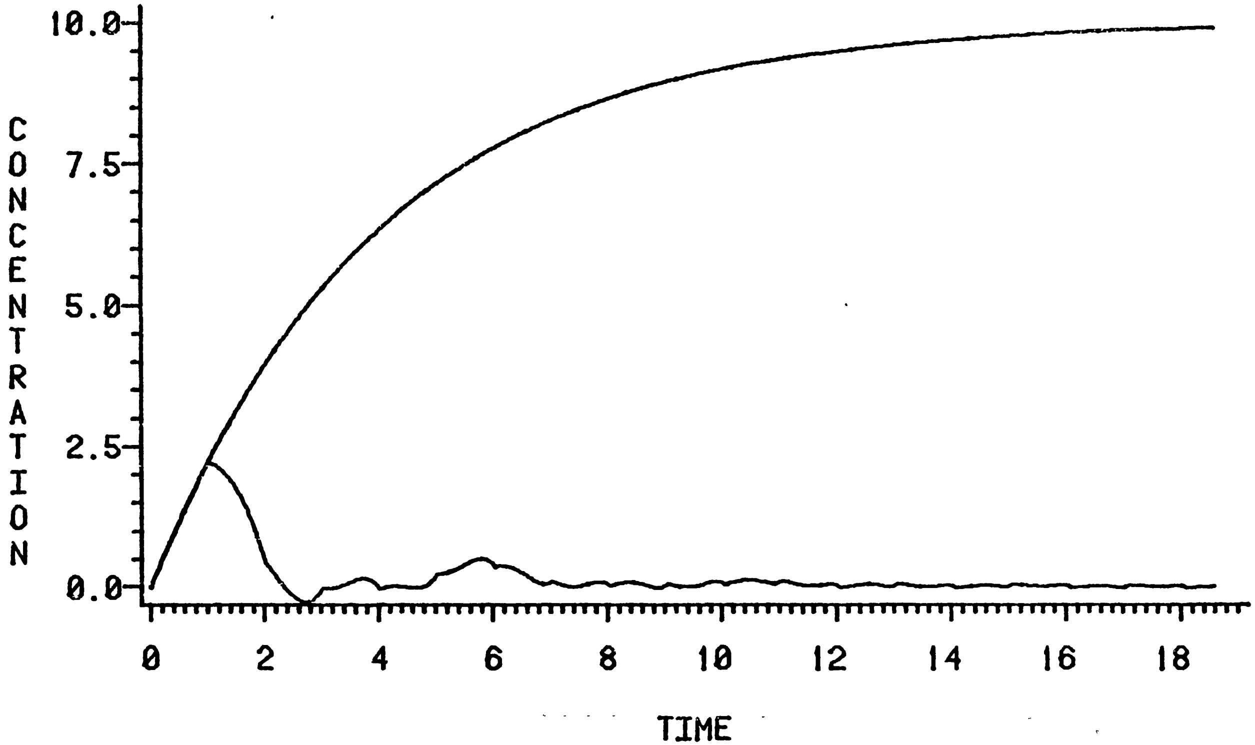


Figure 127. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 4.0$

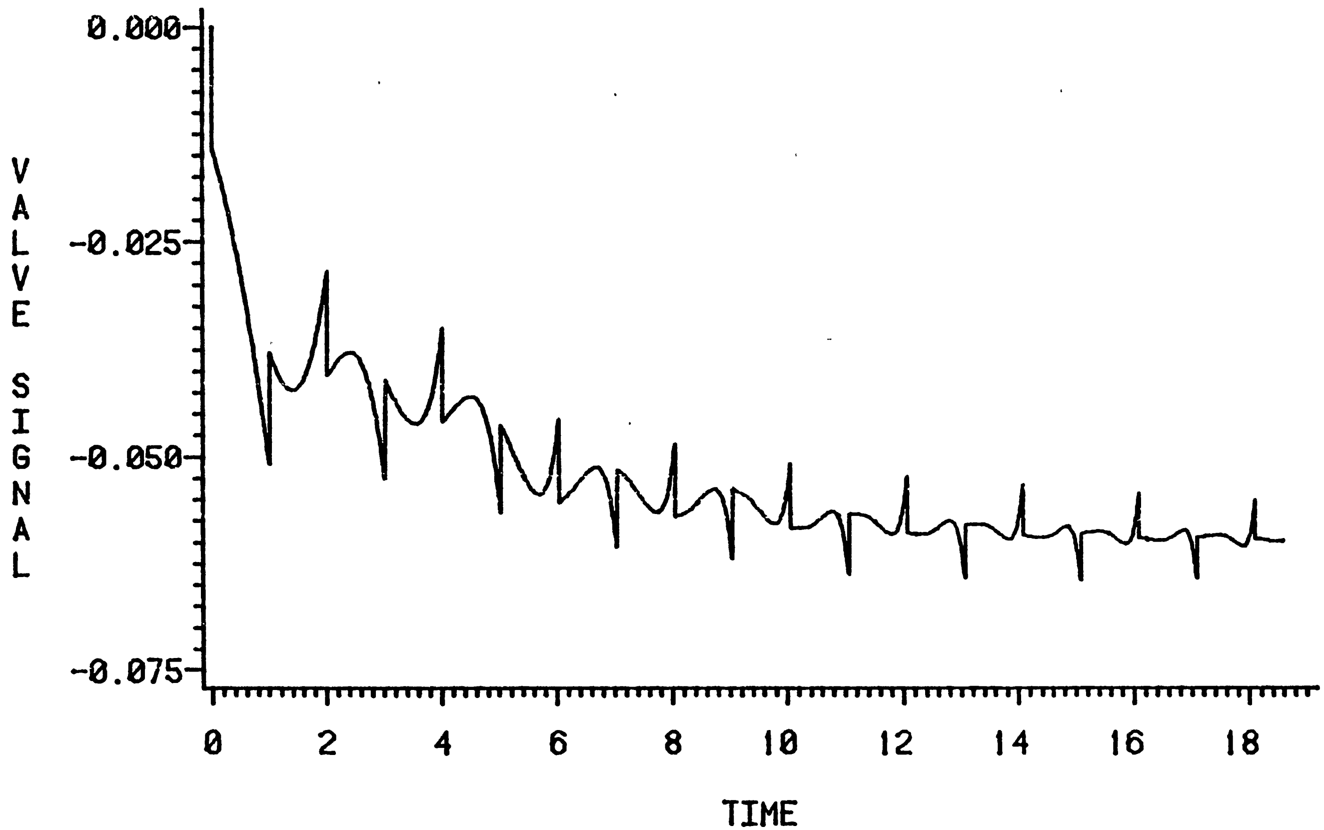


Figure 128. Valve Signal, PID Controller, θ_d/τ_1 = 1.0, τ_2/τ_1 = 4.0

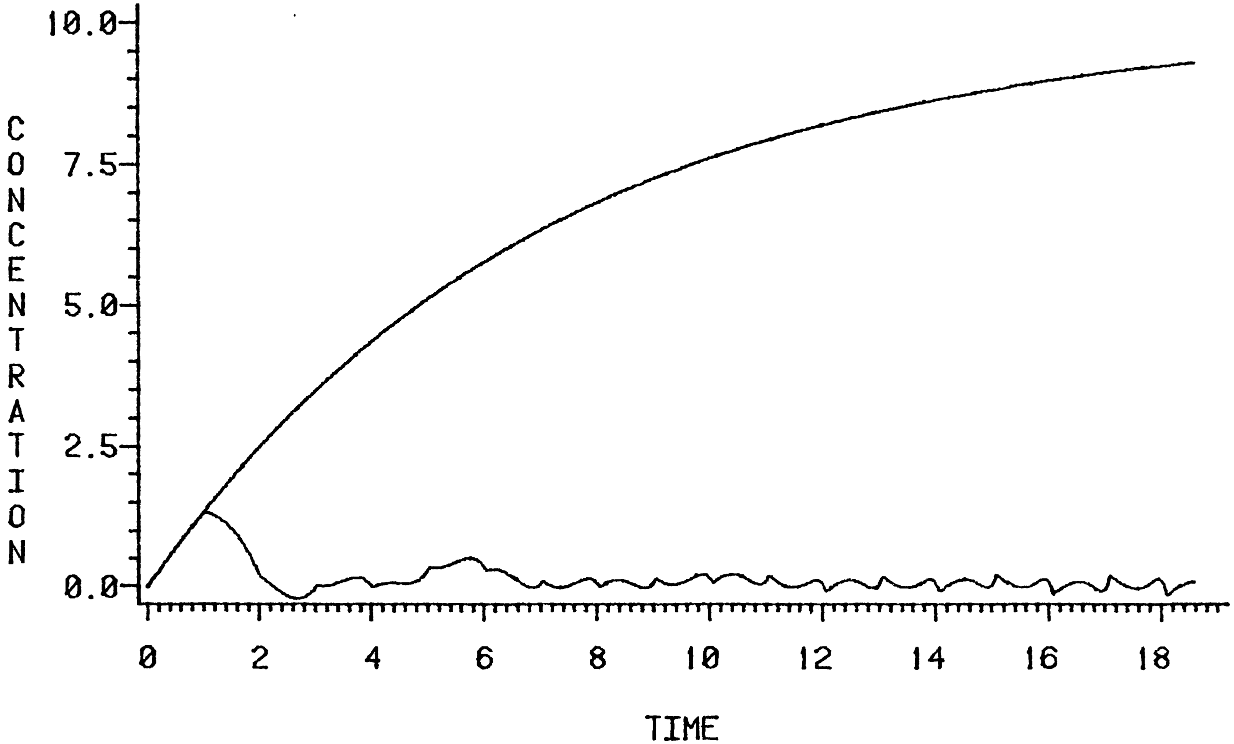


Figure 129. Concentration Variables, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 7.0$

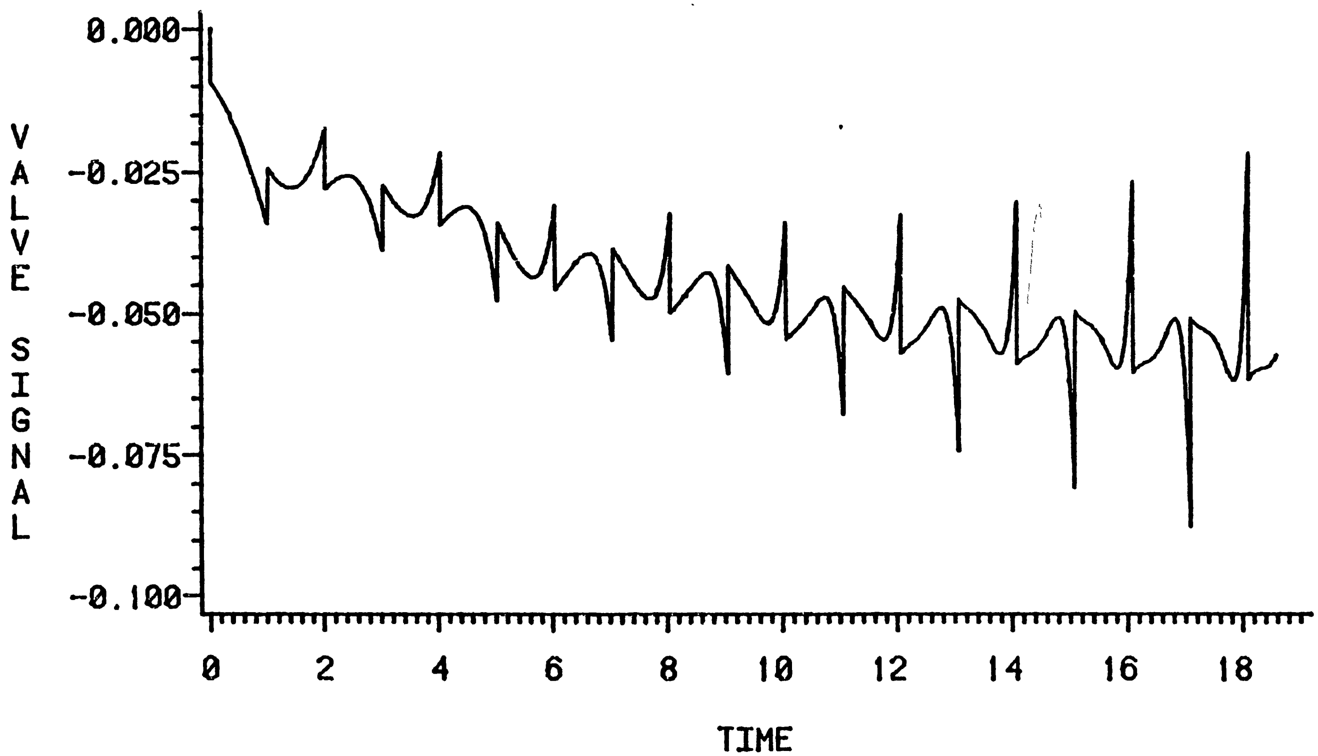


Figure 130. Valve Signal, PID Controller, $\theta_d/\tau_1 = 1.0$, $\tau_2/\tau_1 = 7.0$

APPENDIX D

IAE COMPARISON RATIOS

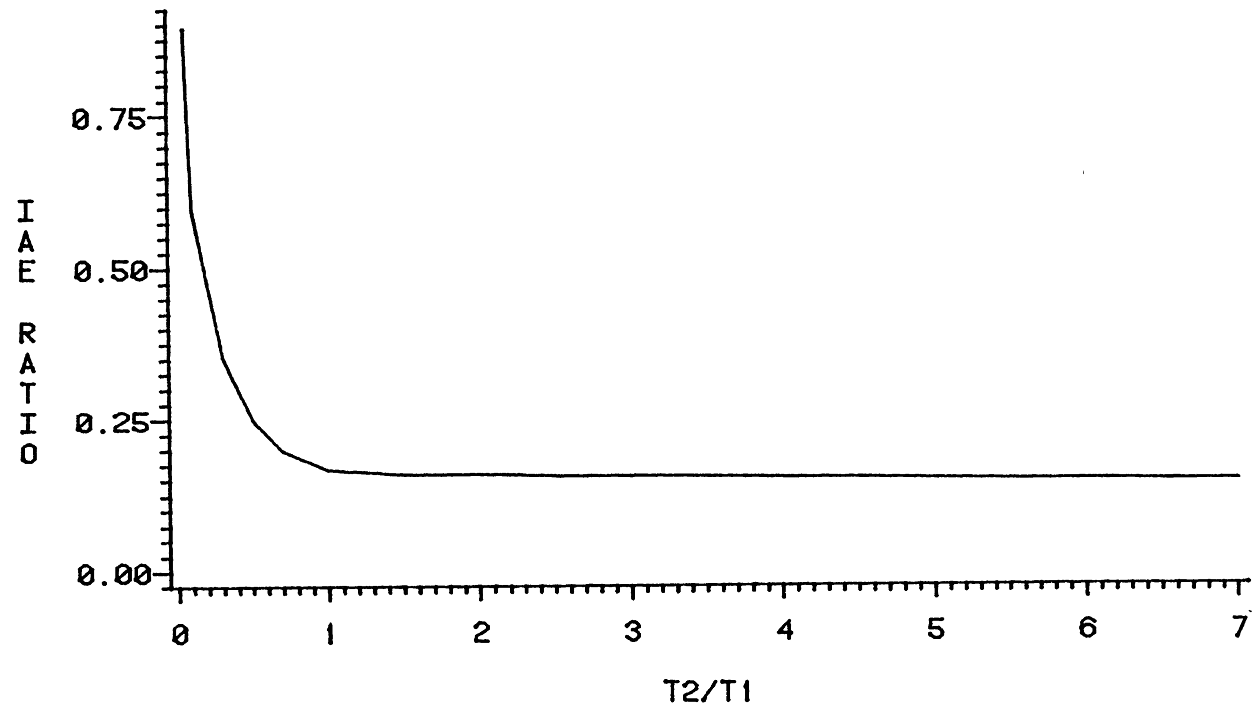


Figure 131. PI Controller, IAE Ratio: Present Method/Rovira Method, $\theta_d/\tau_1 = 0.05$

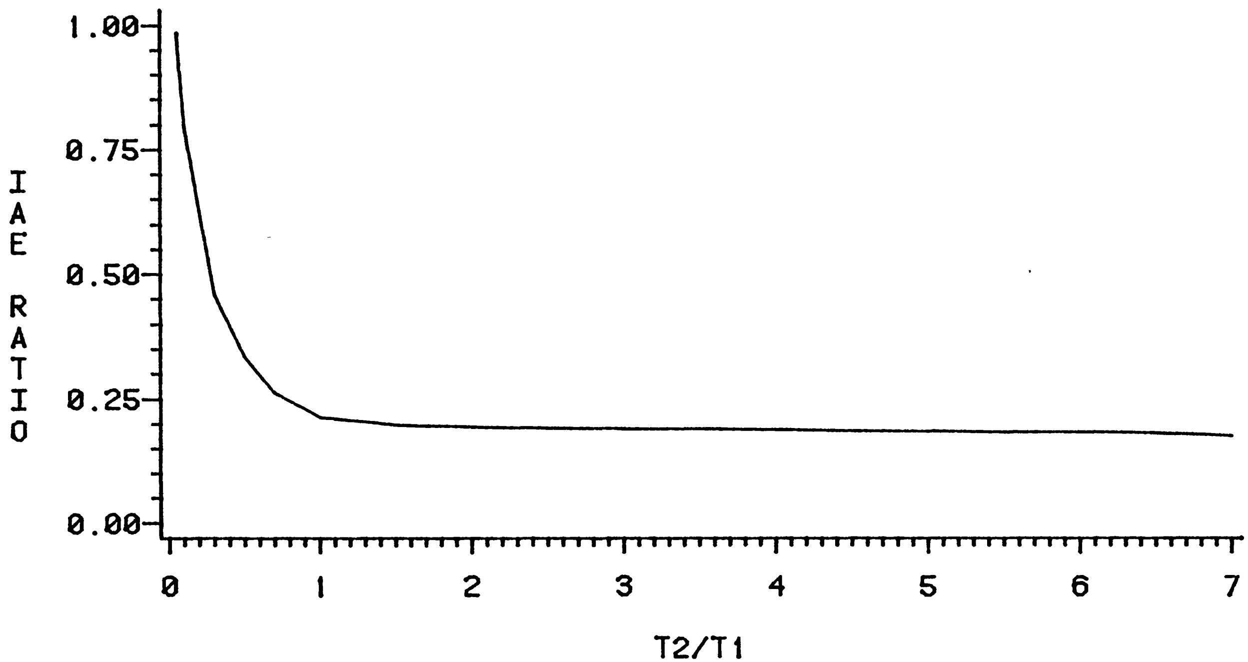


Figure 132. PI Controller, IAE Ratio: Present Method/Rovira Method, $\theta_d/\tau_1 = 0.075$

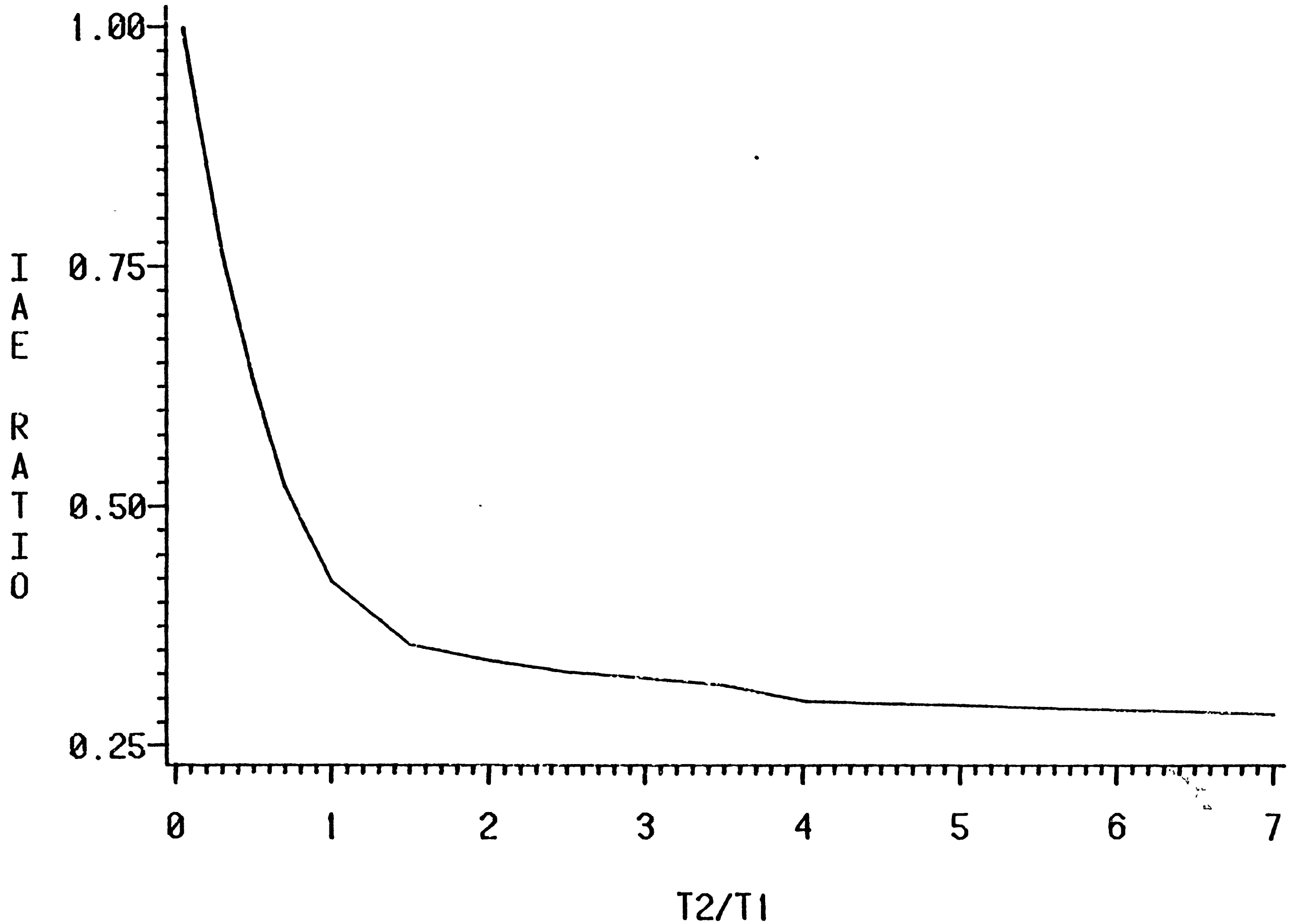


Figure 133. PI Controller, IAE Ratio: Present Method/Rovira Method, $\theta_d/\tau_1 = 0.2$

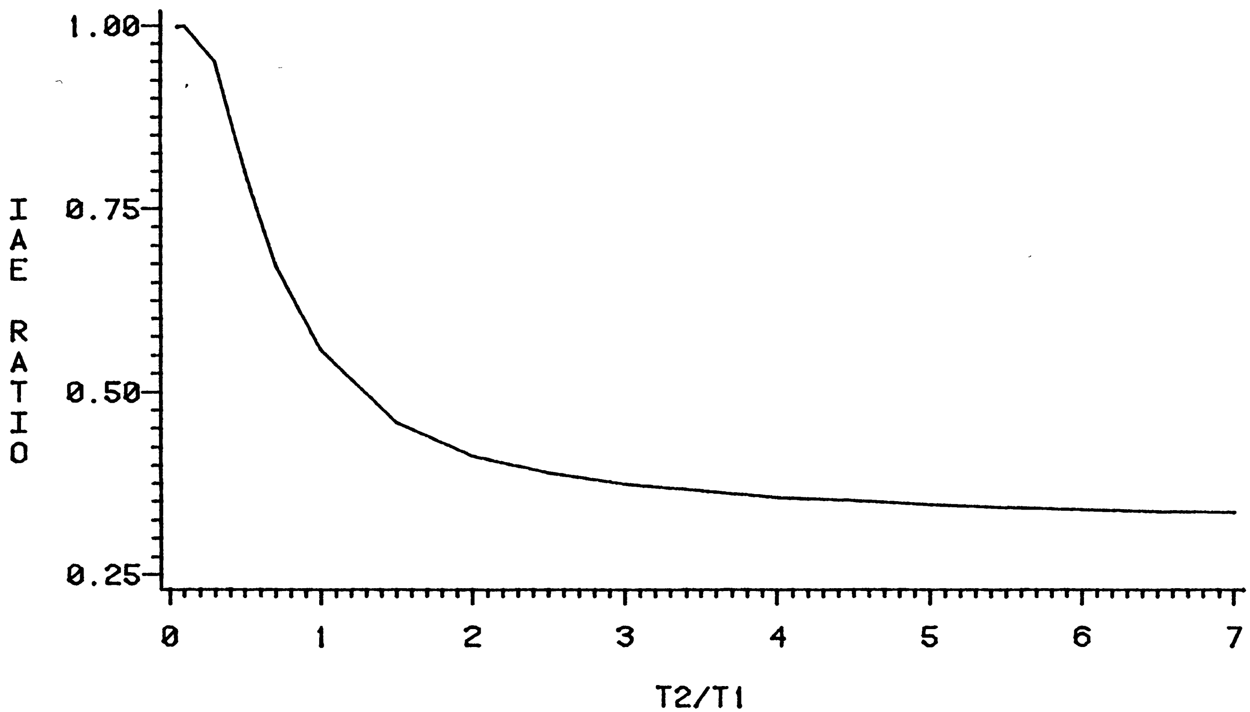


Figure 134. PI Controller, IAE Ratio: Present Method/Rovira Method, $\theta_d/\tau_1 = 0.3$

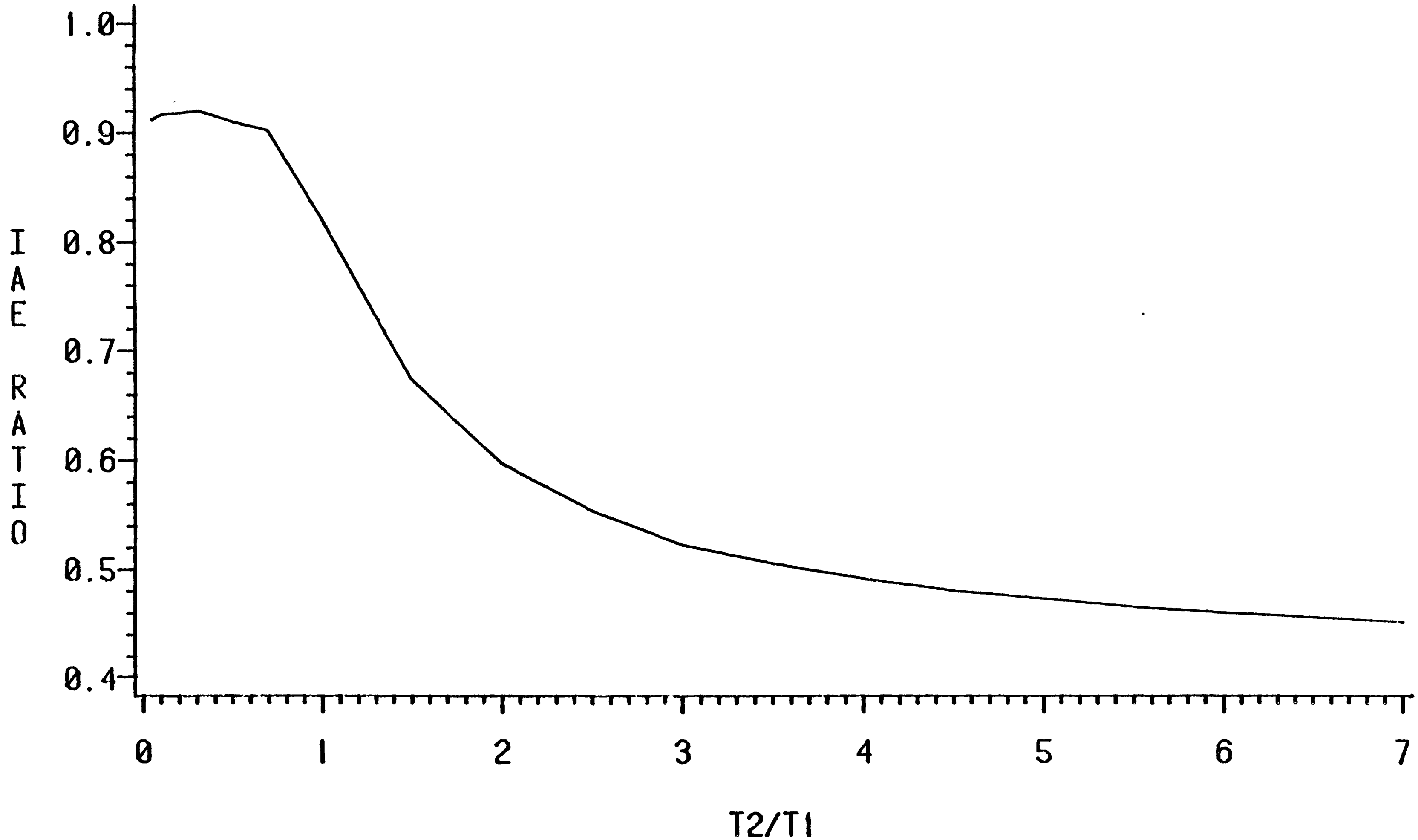


Figure 135. PI Controller, IAE Ratio: Present Method/Rovira Method, θ_d/τ_1 = 0.4

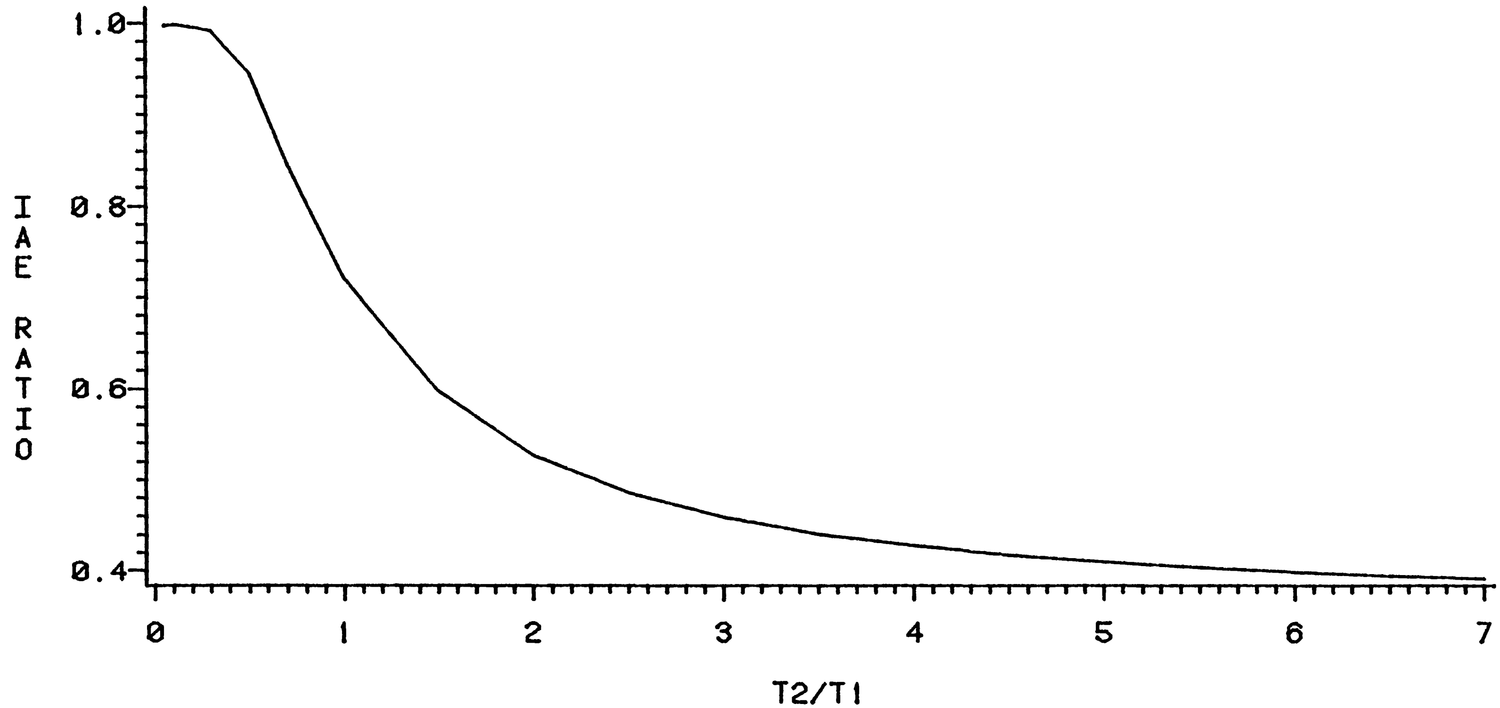


Figure 136. PI Controller, IAE Ratio: Present Method/Rovira Method, θ_d/τ_1 = 0.5

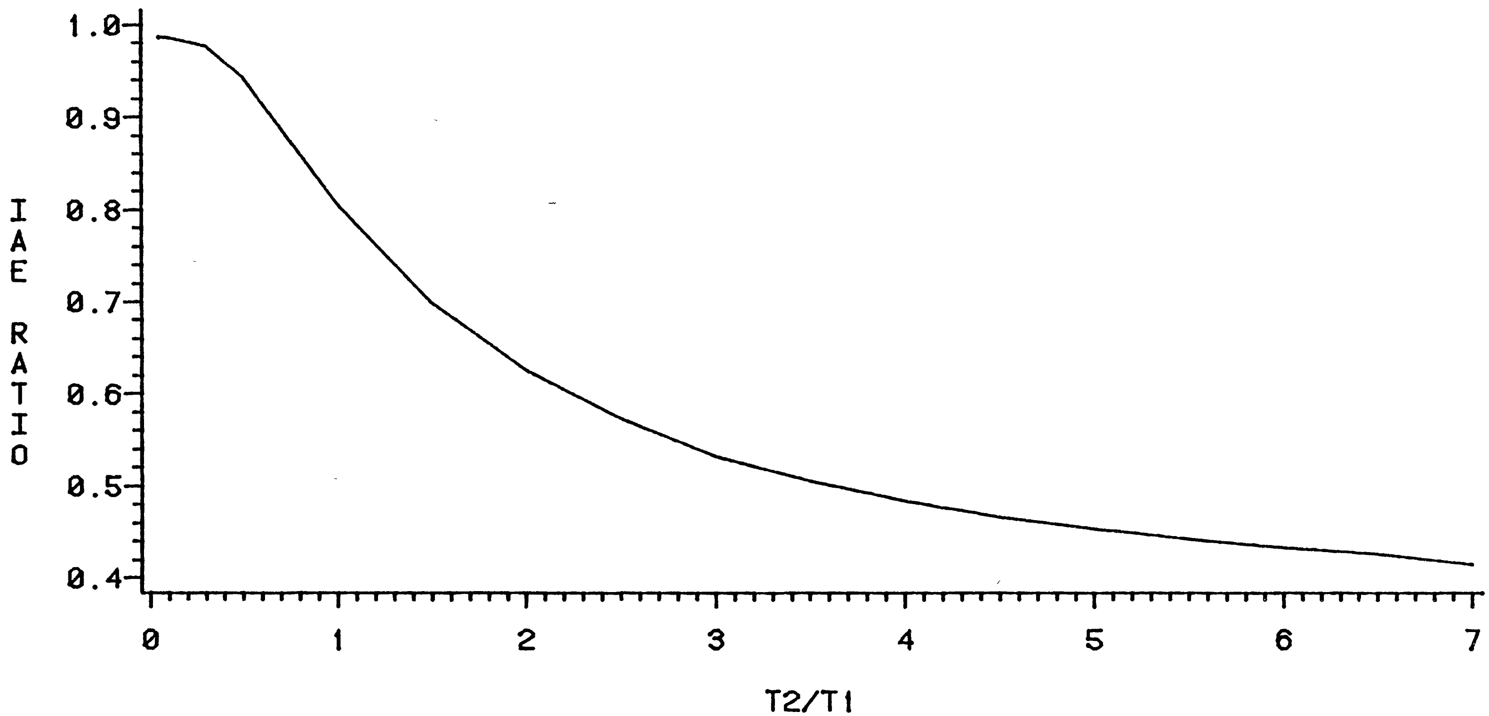


Figure 137. PI Controller, IAE Ratio: Present Method/Rovira Method, θ_d/τ_1 = 1.0

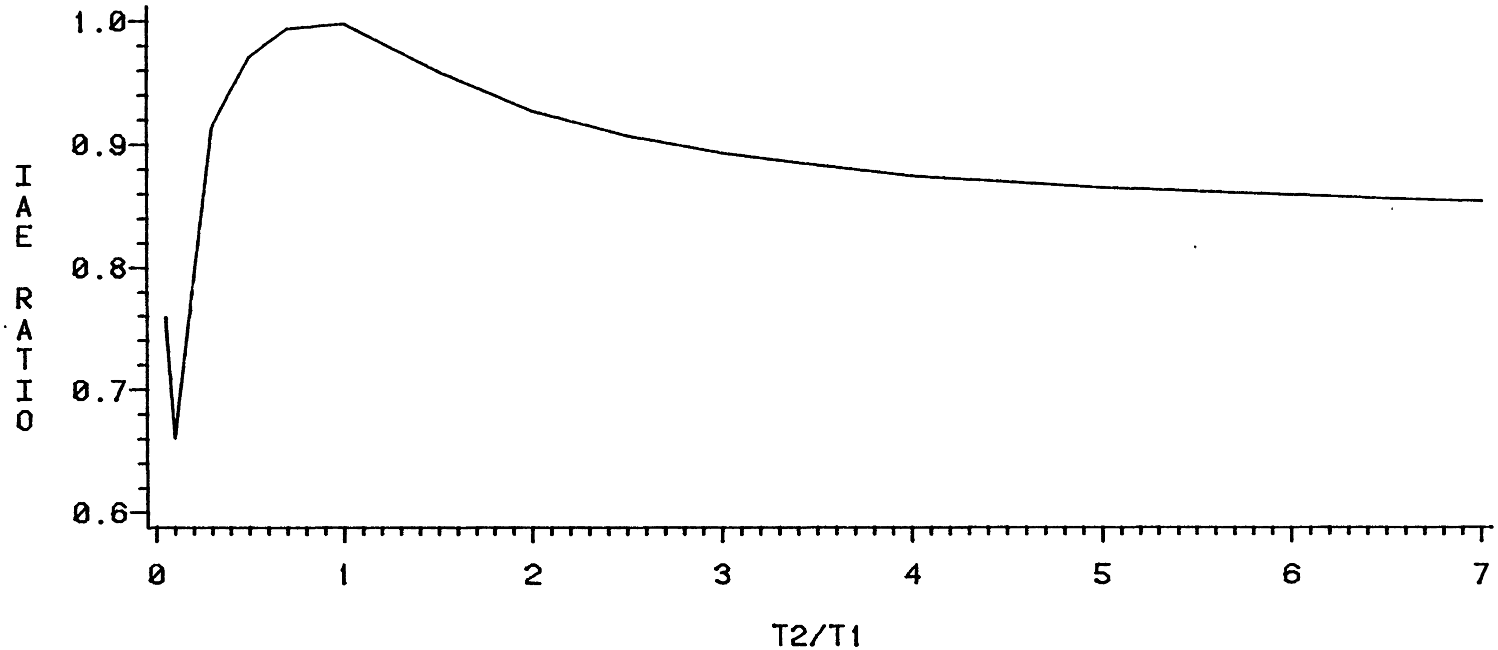


Figure 138. PI Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.05

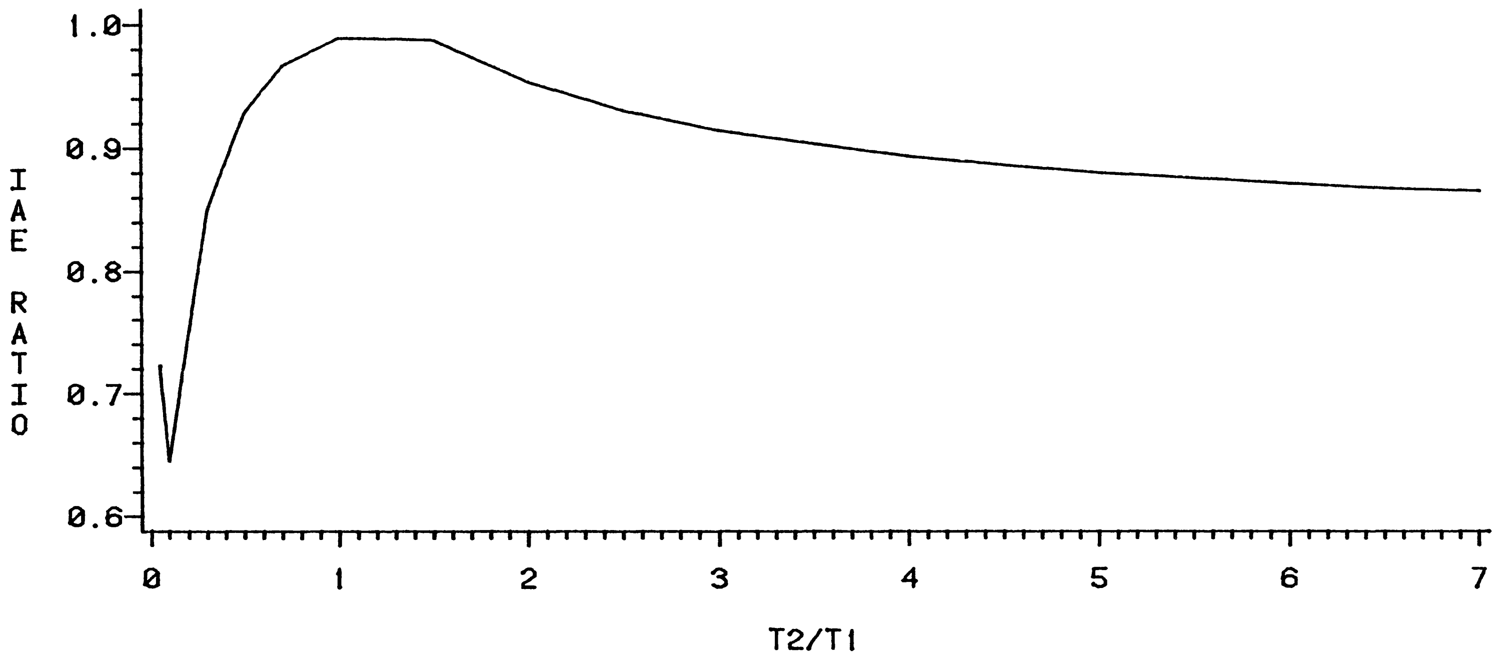


Figure 139. PI Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.075

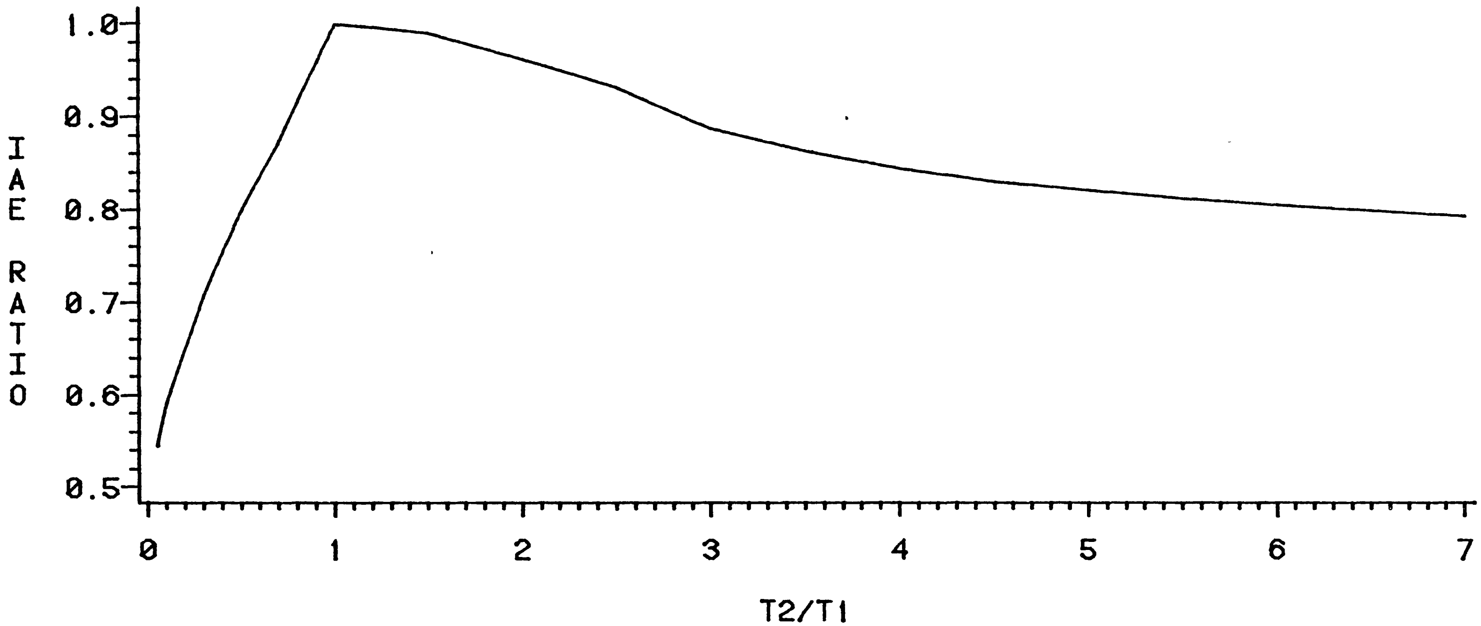


Figure 140. PI Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.2

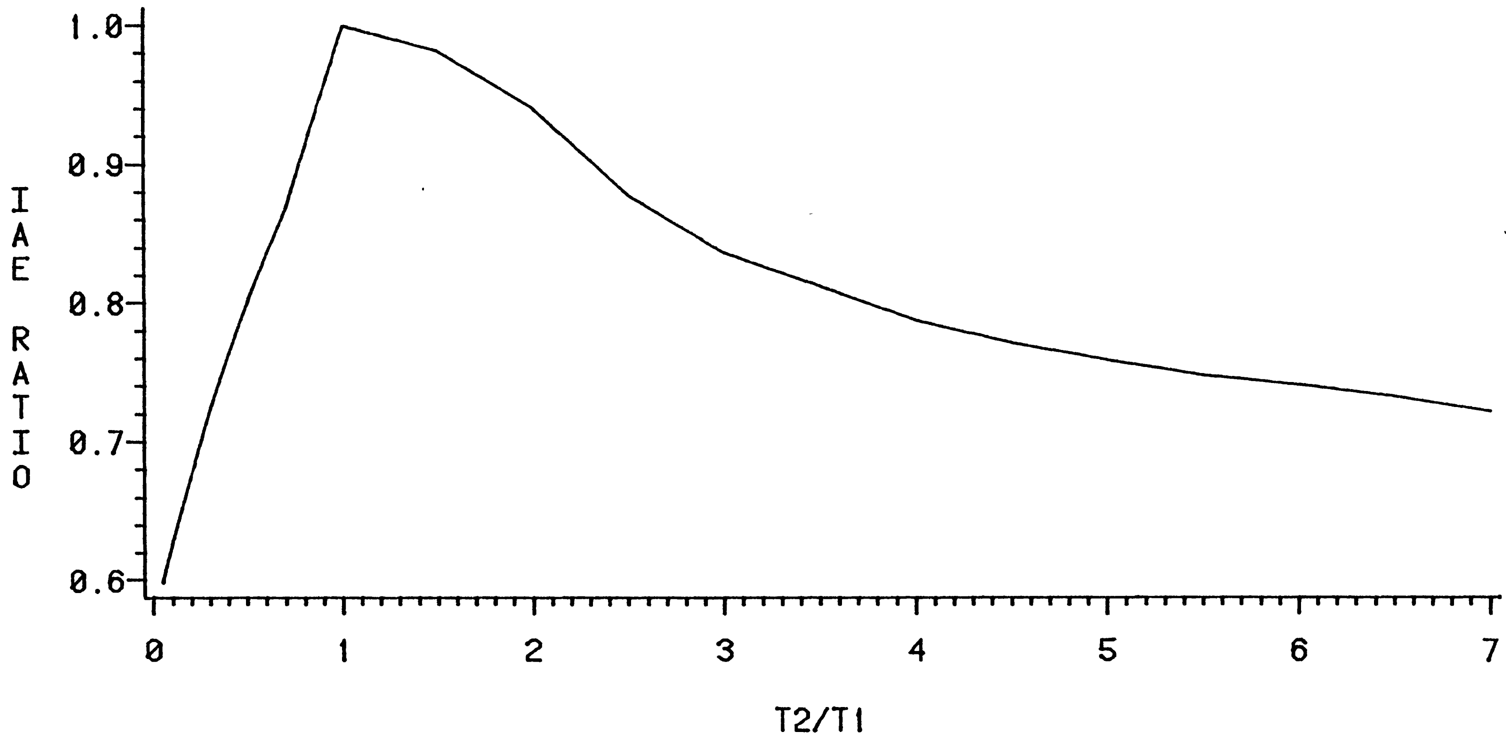


Figure 141. PI Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.3

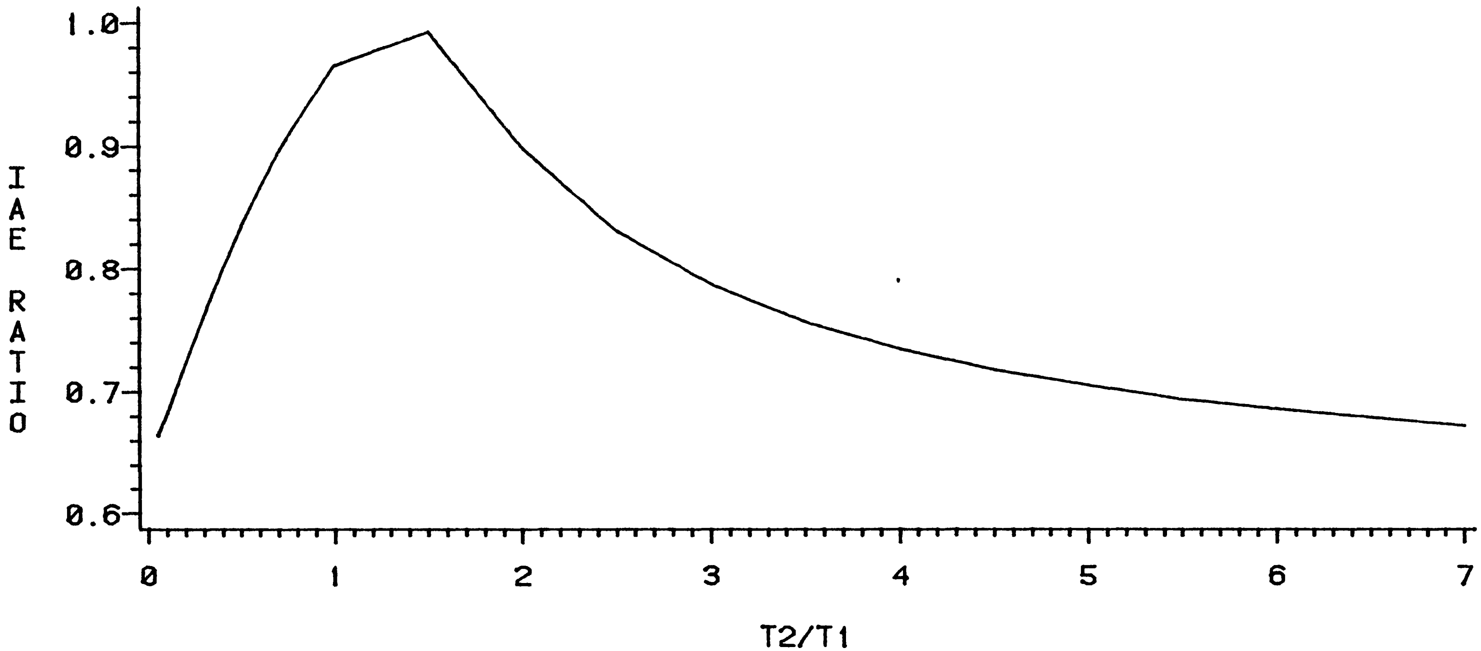


Figure 142. PI Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.4

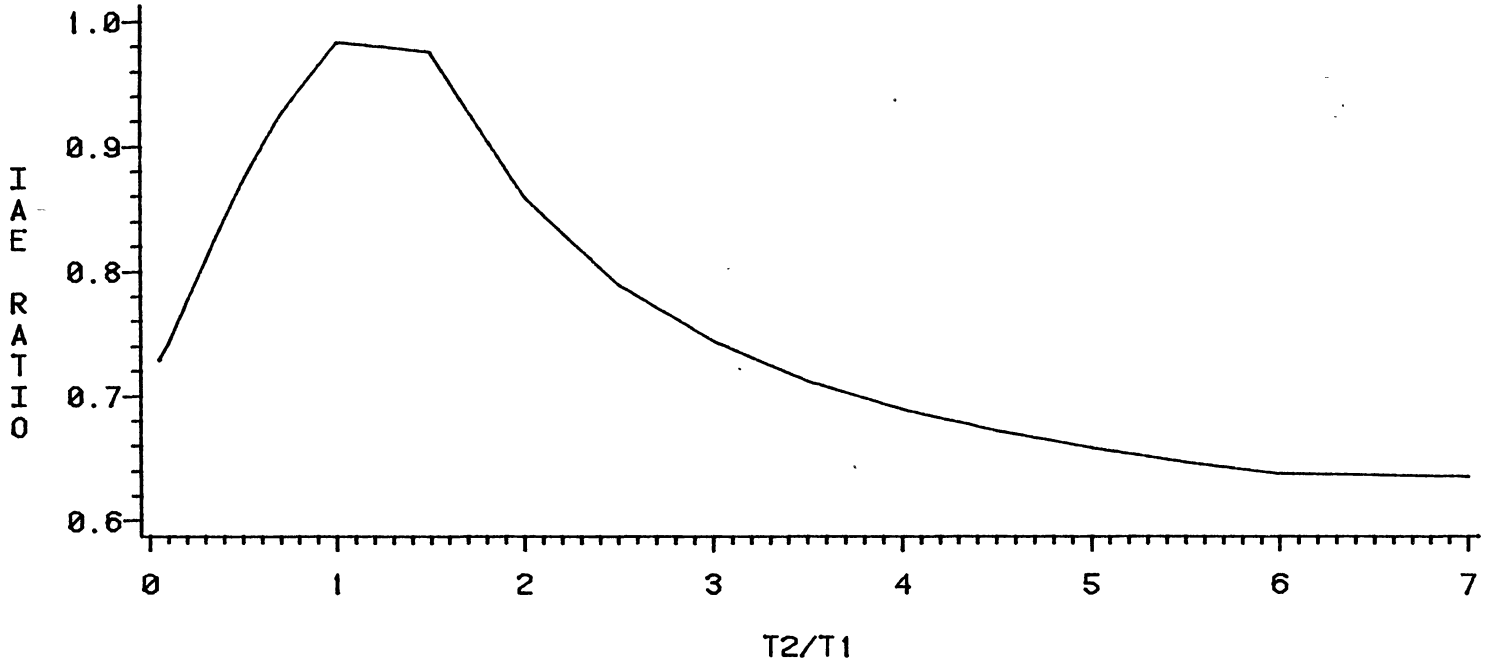


Figure 143. PI Controller, IAE Ratio: Present Method/Lopez Method, $\theta_d/\tau_1 = 0.5$

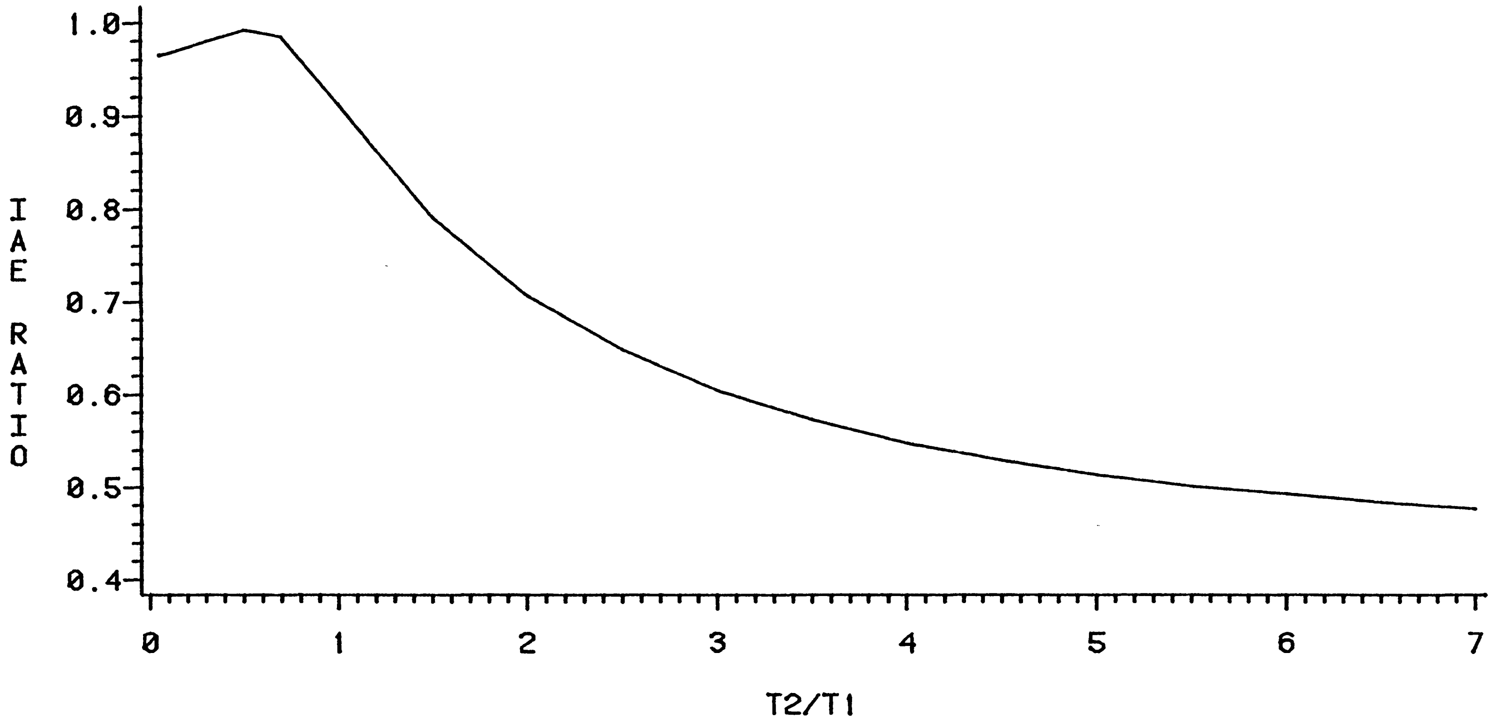


Figure 144. PI Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 1.0

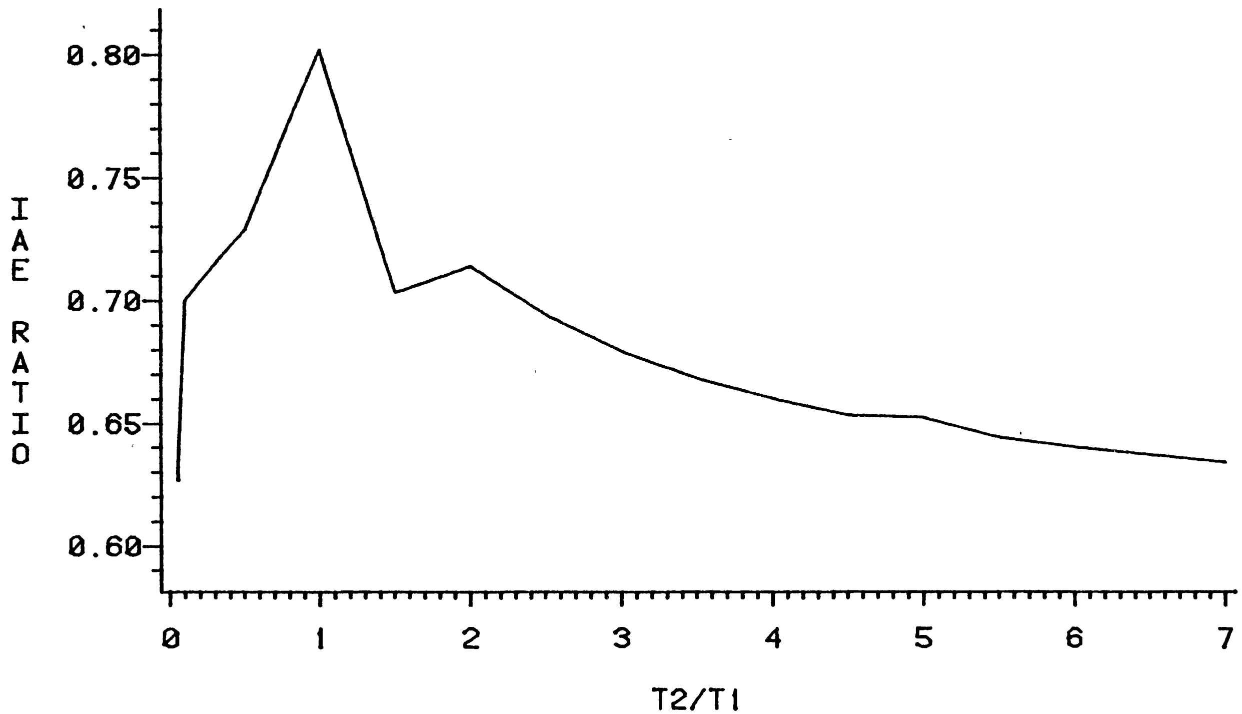


Figure 145. PID Controller, IAE Ratio: Present Method/Lopez Method, $\theta_d/\tau_1 = 0.05$

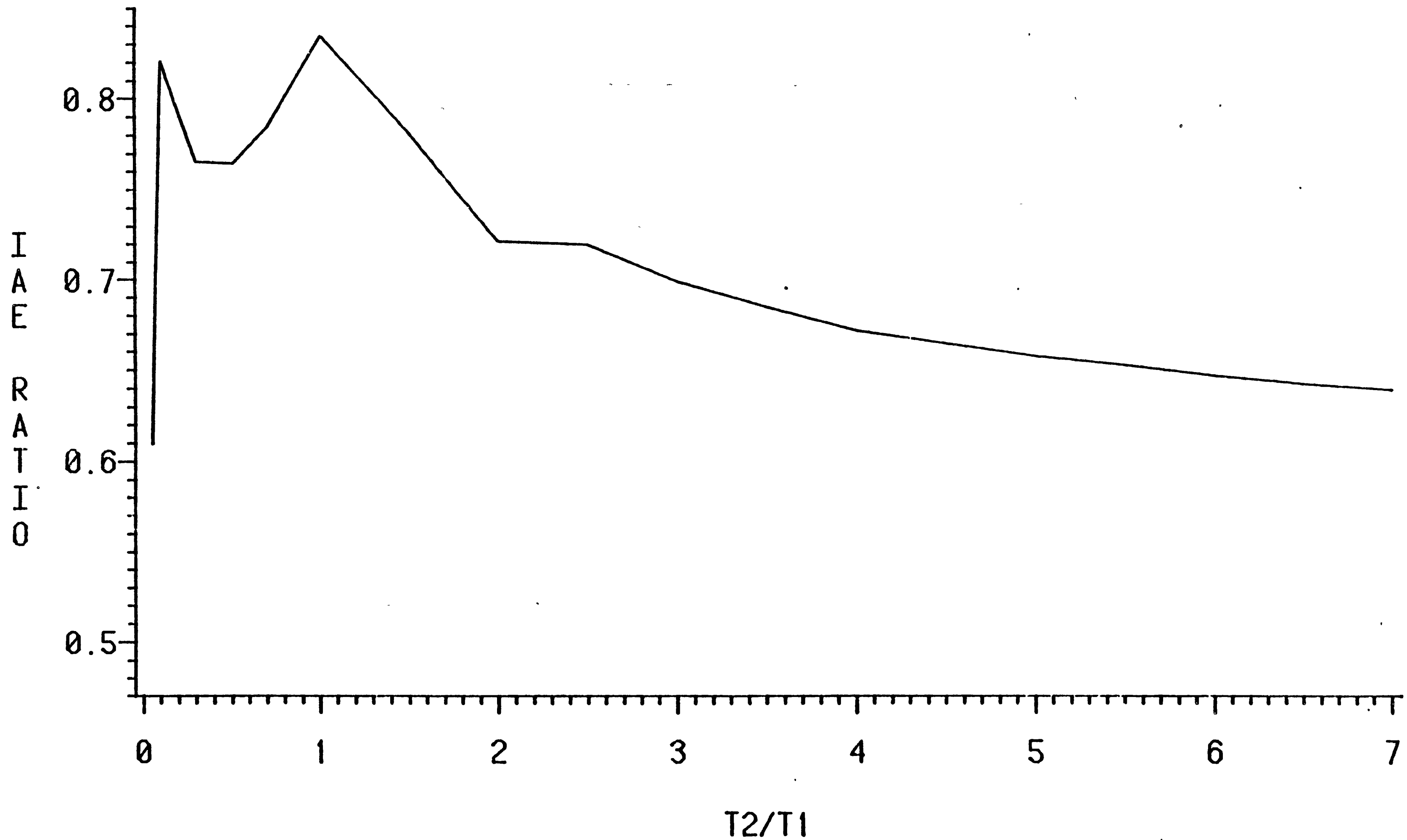


Figure 146. PID Controller, IAE Ratio: Present Method/Lopez Method, $\theta_d/\tau_1 = 0.075$

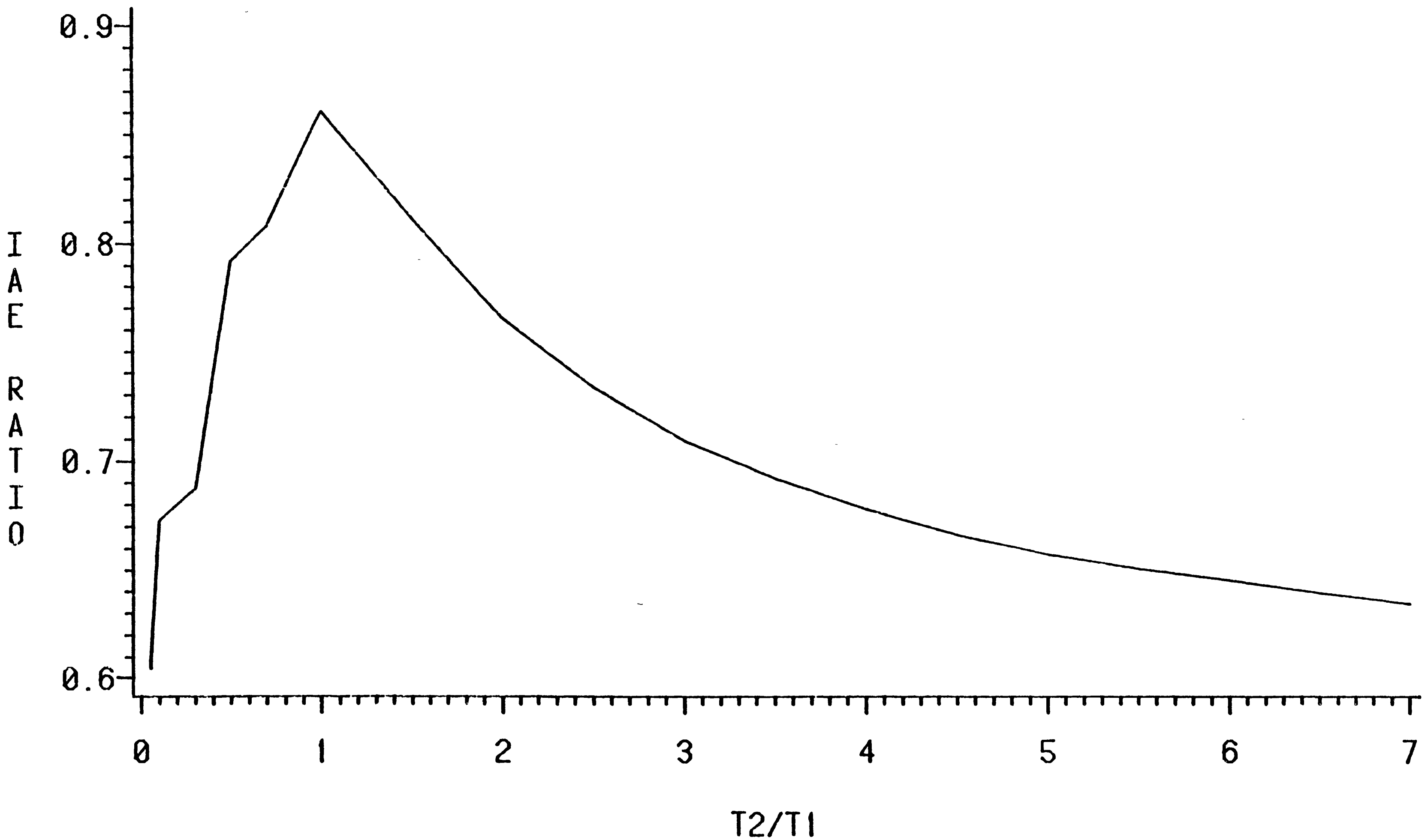


Figure 147. PID Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.1

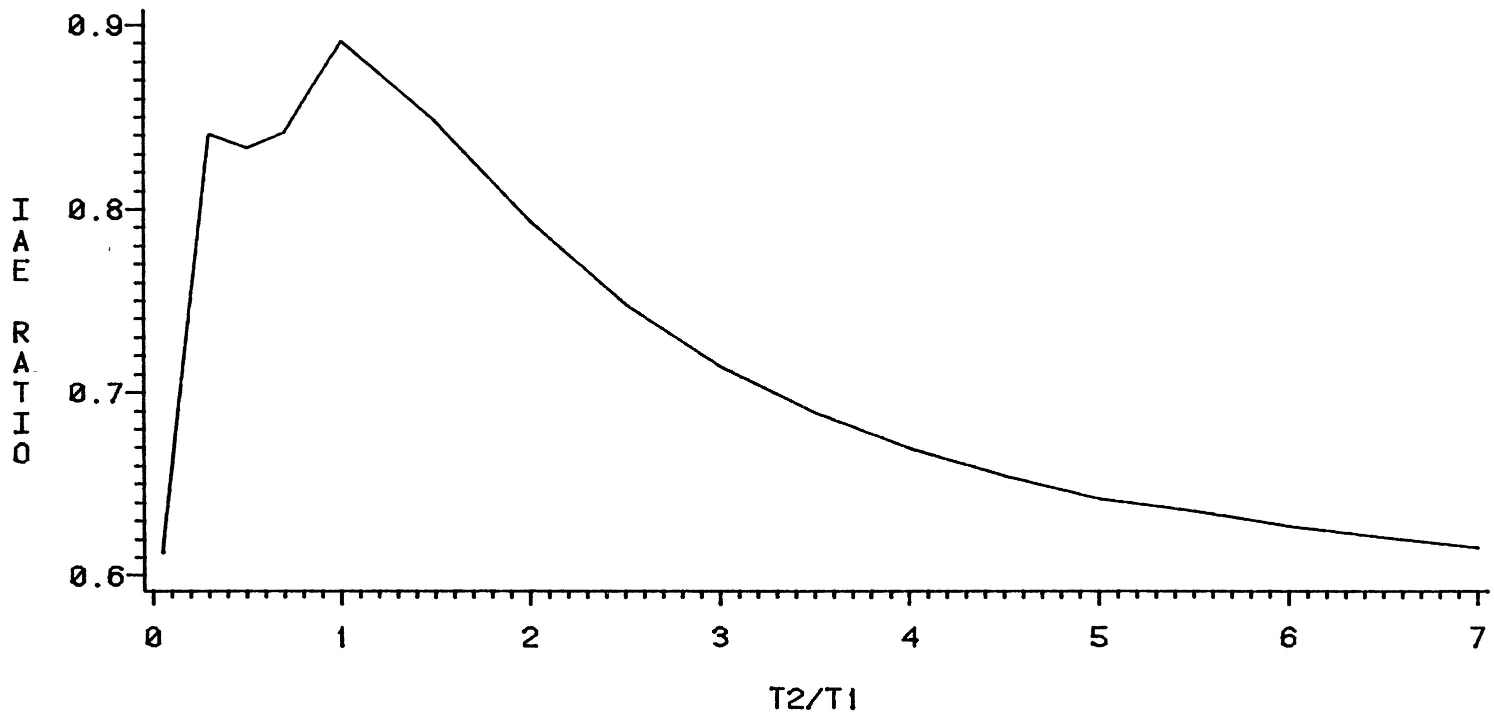


Figure 148. PID Controller, IAE Ratio: Present Method/Lopez Method, $\theta_d/\tau_1 = 0.15$

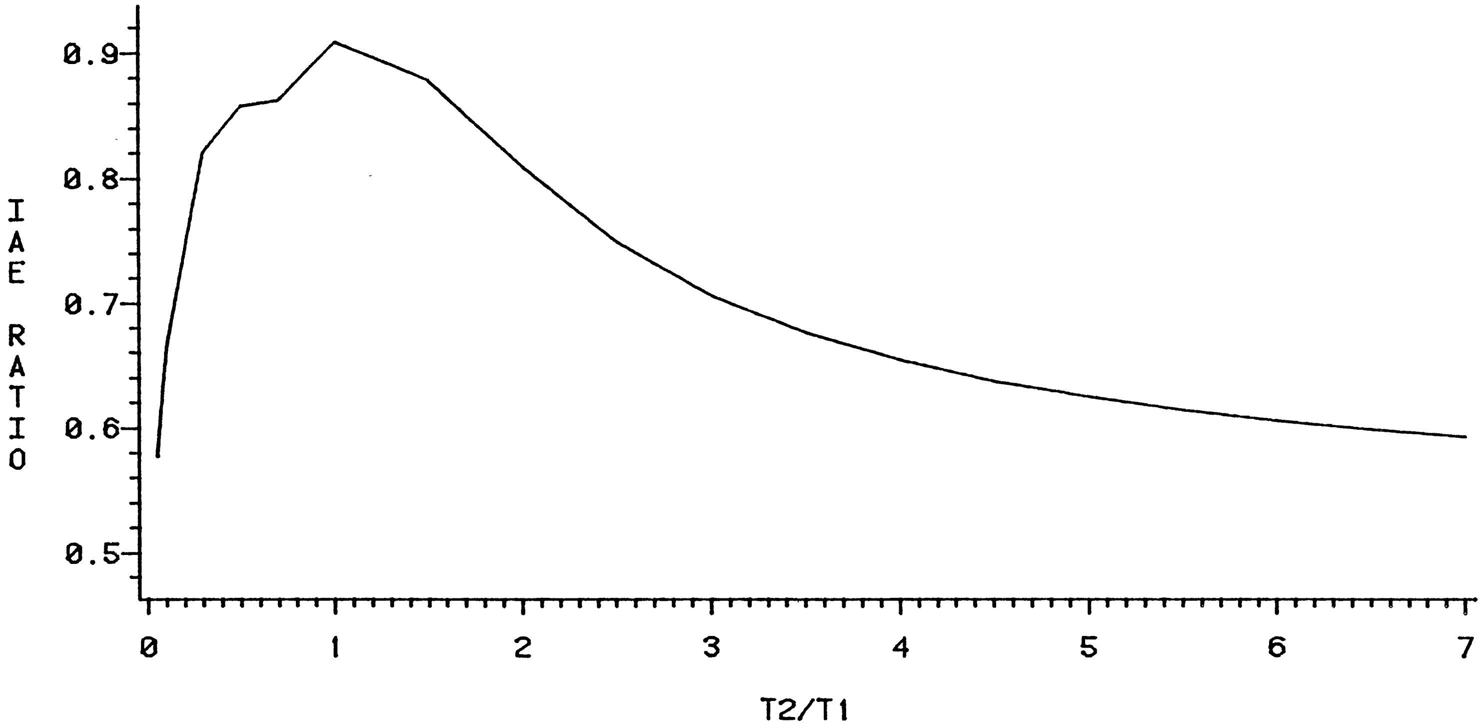


Figure 149. PID Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.2

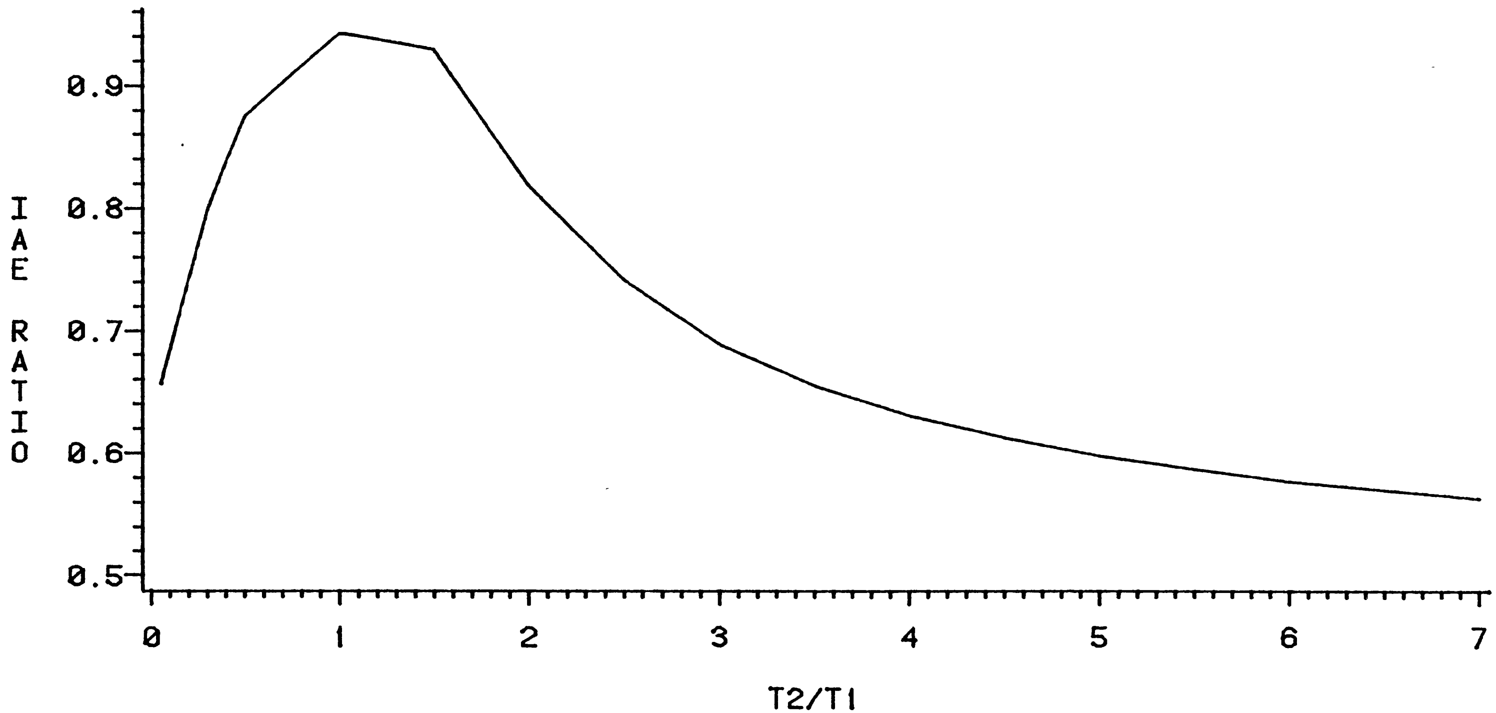


Figure 150. PID Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.3

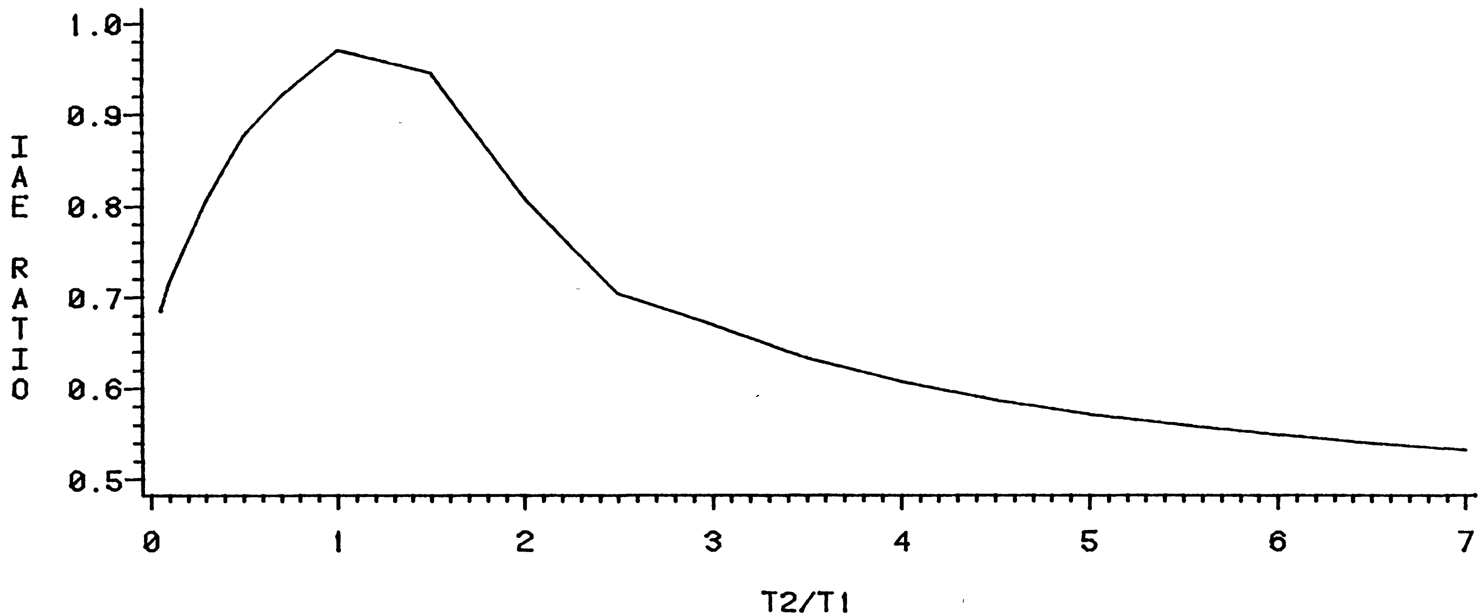


Figure 151. PID Controller, IAE Ratio: Present Method/Lopez Method, $\theta_d/\tau_1 = 0.4$

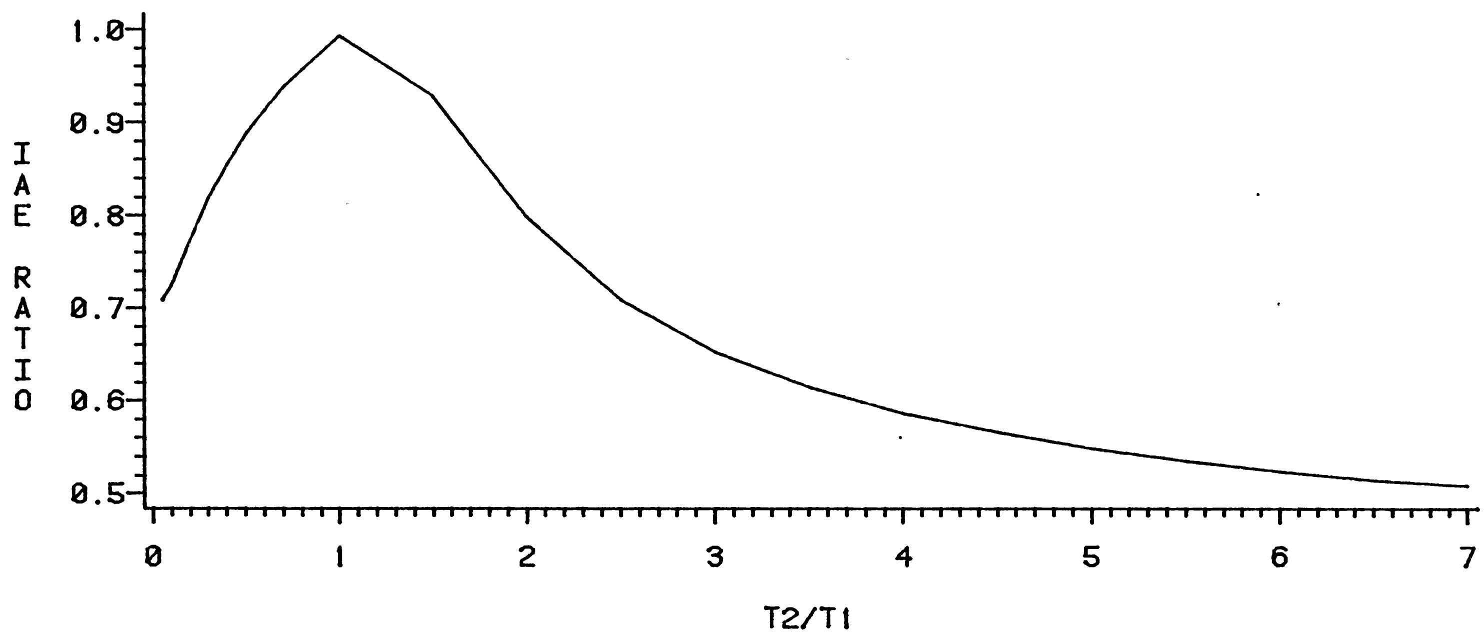


Figure 152. PID Controller, IAE Ratio: Present Method/Lopez Method, $\theta_d/\tau_1 = 0.5$

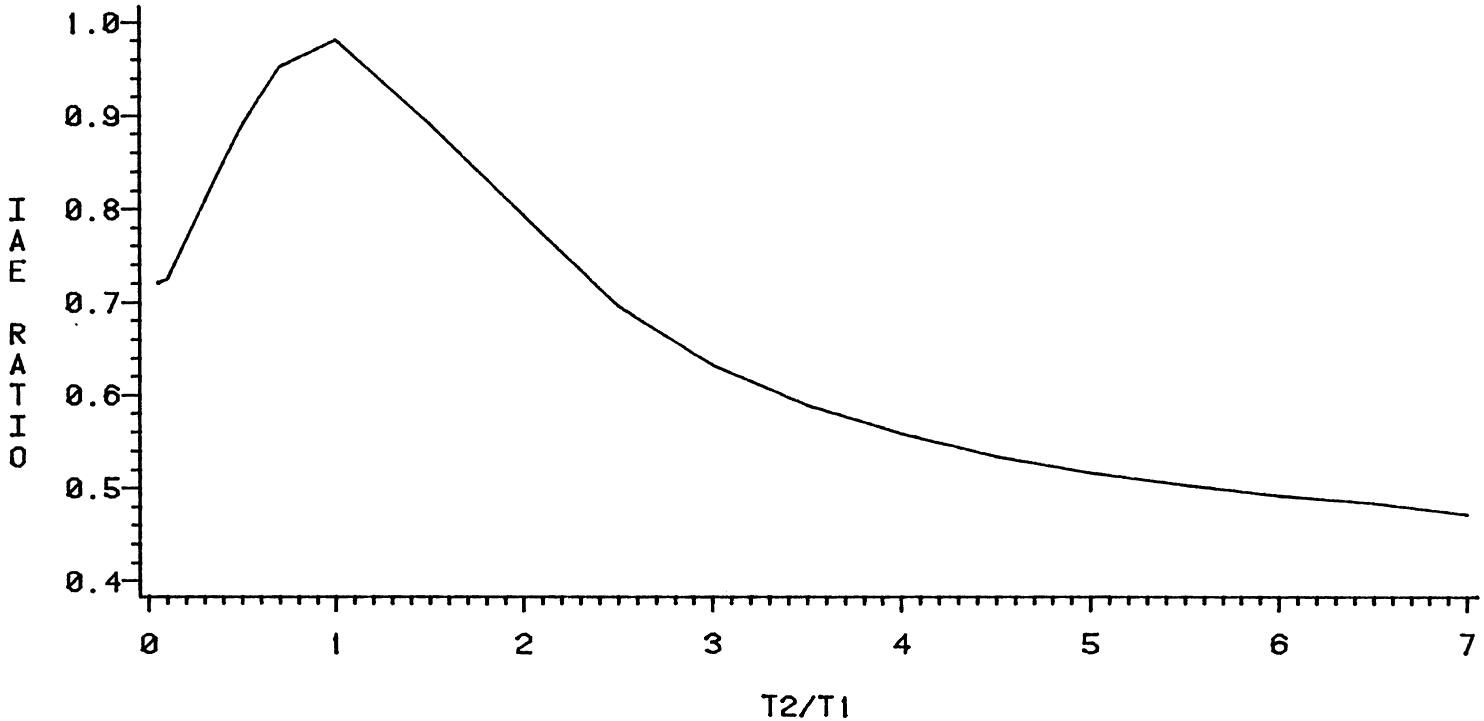


Figure 153. PID Controller, IAE Ratio: Present Method/Lopez Method, θ_d/τ_1 = 0.75

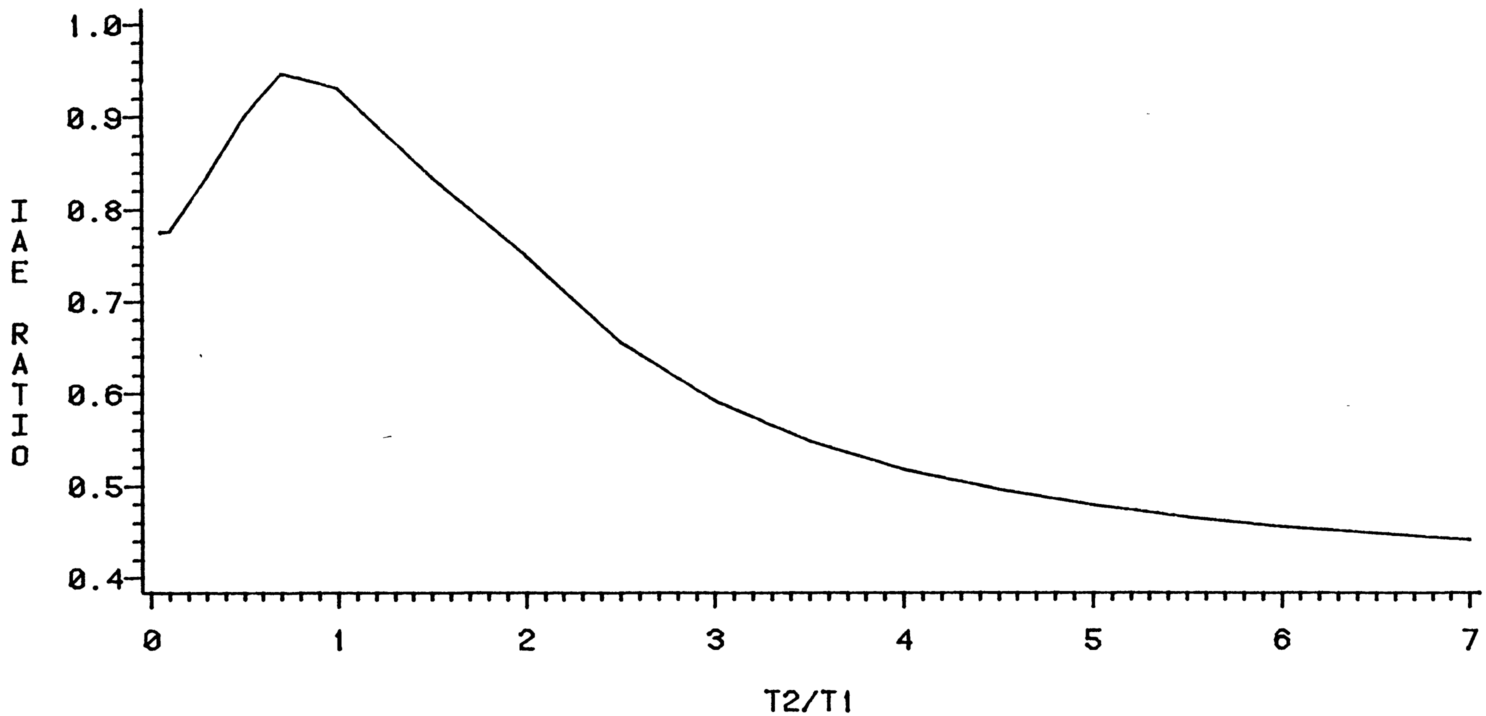


Figure 154. PID Controller, IAE Ratio: Present Method/Lopez Method, $\theta_d/\tau_1 = 1.0$

VITA 2

Siripol Kosinsani

Candidate for the Degree of

Doctor of Philosophy

Thesis: EFFECT OF DISTURBANCE DYNAMIC CHARACTERISTICS ON OPTIMUM PID CONTROLLER TUNING CONSTANTS

Major Field: General Engineering

Biographical:

Personal Data: Born in Cha-Chern-Sao, Thailand, August 16, 1950, the son of Thamp and Prapi Kosinsani.

Education: Graduated from Saint Louis School, Cha-Chern-Sao, Thailand, in March 1969; received Bachelor of Science degree in Agricultural Mechanization from Kasetsart University, Bangkok, Thailand in April, 1973; received Master of Agriculture degree from Mississippi State University in December, 1976; performed graduate studies at Ohio State University between Spring Quarter 1979 to Autumn Quarter 1980; completed the requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1985.

Professional Experience: Teaching Instructor, Department of Agricultural Mechanization, North-Eastern Technological Institute, Nakorn-raj-srima, Thailand, May 1973 - October 1973; Engineer, Agricultural Engineering Division, Department of Agriculture, Bangkok, Thailand, November 1973 - present.